KeTop T40

Handheld Terminal
Serial Gateway Coupling

User's Manual V 1.1

KEBA®

Automation by innovation.

Notes on This Manual

At various points in this manual you will see notes and precautionary warnings regarding possible hazards. The meaning of the symbols used is explained below.

⚠ DANGER

- **DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.**

⚠ WARNING

- **WARNING indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.**

⚠ CAUTION

- **CAUTION indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.**

CAUTION

- **CAUTION used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property injury.**

This symbol reminds you of the possible consequences of touching electrostatically sensitive components.

Note

Notes on use of equipment and useful practical tips are identified by the "Notice" symbol. Notices do not contain any information that draws attention to potentially dangerous or harmful functions.

Document: version 1.1 / material no.: 62010
Filename: t40_gateway_en.doc, last saving on: 9. 12. 2004

KEBA AG, Postfach 111, Gewerbepark Urfahr, A-4041 Linz
Tel.: ++43 / 732 / 70 90-0, Fax: ++43 / 732 / 73 09 10, E-Mail: keba@keba.com, www.keba.com

KEBA GmbH, Ulmer Straße 123, D-73037 Göppingen
Tel.: ++49 / 7161 / 97 41-0*, Fax: ++49 / 7161 / 97 41-40

KEBA Corp., 100 West Big Beaver Road, Troy, MI 48084
Tel. ++1 / 248 / 526 - 0561, Fax: ++1 / 248 / 526 - 0562, E-Mail: schr@us.keba.com

History

Modification from / to	Date	Modified pages	Description	Author
V1.0	8-2002	-	First release	sam
V1.0/V1.1	12-2004	all	New Layout	sam

Contents

1 Introduction

This document is a supplement to the User's Manual "KeTop T40 Handheld Terminal - General Information" and describes the serial gateway protocol.

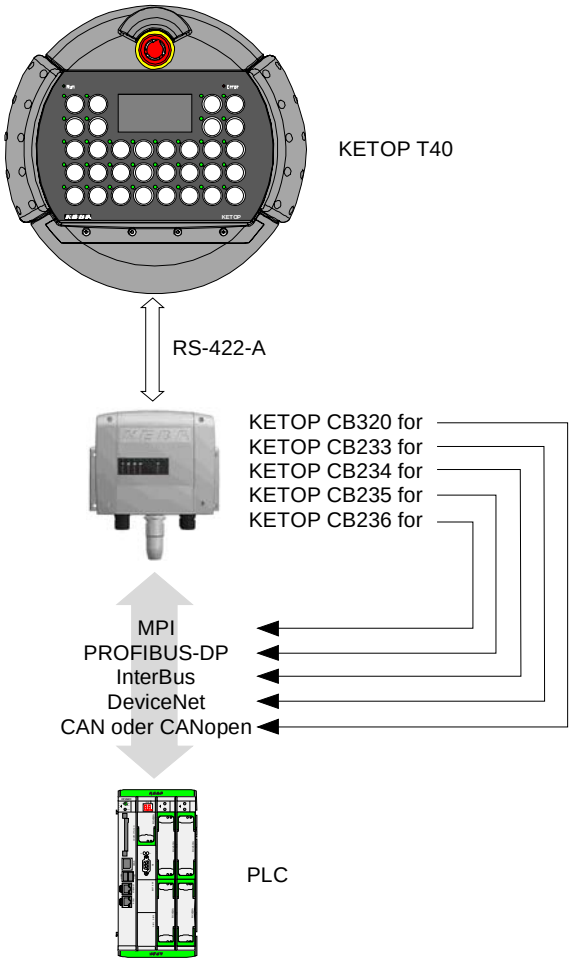
For a detailed description of the programming, the special functions and the key labelling of the HT, please refer to the User's Manual "KeTop T40 Handheld Terminal - General Information".

2 Connection

The serial gateway box can be connected to the handheld terminal via the serial interface COM2. By means of DIP switches, this interface must be configured as:

COM2: RS-422-A

See also User's Manual "KeTop T40 Handheld Terminal - General Information", chapter "Connection".



Connection of HT via gateway box

3 Programming the Handheld Terminal

Programming Software

For setting the device configuration and creating texts, we deliver a programming software which is executable under Windows.

The programming of the handheld terminal is described in detail (e.g. functions for editing the keypad assignment and for loading the program) in the User's Manual "KeTop T40 Handheld Terminal - General Information" and applies in general also to the serial gateway coupling.
Therefore the following chapter only describes the specific details of the gateway coupling.

Configuration

Selection of protocol

After selection of the protocol **„Seriell Gateway MMI-COM“** the following parameters must be set:

Interface	Selection of required interface.
Baudrate	Selection of baud rate required for transmission: 9600, 19200, 38400, 57600, 115200 Baud.
Parity	The setting of priority is NO and cannot be changed.
Data bit	The number of data bits is 8 and cannot be changed.
Stop bit	The number of stop bits is 1 and cannot be changed.

Numbering of variables

For the serial use of the HT the following **restriction** must be taken into account in the text editing mode:

User variables may be defined only from variable number 100 on. The range from 1 to 99 is reserved for the system variables (see also chapter "HT System Variables").

EXCEPTION: string variables (numbering from 0 to 255)

4 HT Power-Up Phase after Turning On

The handheld terminal carries out a self-test after turning on. For details on the test steps, refer to the User's Manual "KeTop T40 Handheld Terminal - General Information".

- The first part of the test is identical for all KeTop T40 couplings and therefore described in the User's Manual "KeTop T40 Handheld Terminal - General Information".
- Then the configuration data are loaded.

The following message is displayed:

```
Serial Gateway      Vx.x

      Initializing...
```

In case of an error, the handheld terminal remains in this condition.

- After the configuration data have been loaded successfully, the following mask appears on the display:

```
Serial Gateway      Vx.x
Profile: MMI-COM
Channel: COMn, mmmmm
         ooooo,NO,8,1
```

x	program version
n	number of interface port
m	type of interface (RS232 or RSxx2 for RS232/RS422)
o	baud rate (9600, 19200, 38400, 57600, 115200 Baud)

- After approx. 1 second, the display indicates:

```
Serial Gateway      Vx.x

        Wait for
   gateway connection
```

After successful establishment of the communication with the gateway box and after successful loading of the serial configuration data, the interface parameters will be displayed again.

This message remains displayed until it is overwritten by a text activated by the PLC.

5 Protocol

The coding of the telegrams is based on the MMI-COM profile which was specified for operating and display panels used on the INTERBUS.

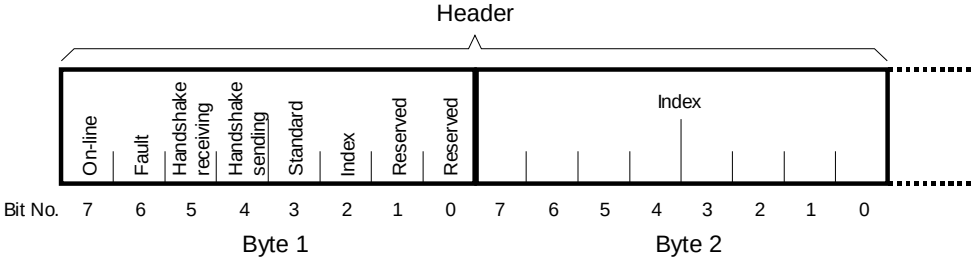
MMI-COM

MMI-COM signifies „Man Machine Interface-Communication“. This profile definition standardizes the communication so that every station interprets the data in the same way. Functions standardize some essential parameters of an MMI device.

The communication model of MMI-COM is based on the fact that data are sent to objects or by objects. The objects are addressed with a number (index). In general, the connection with the index specifies how the data are to be interpreted. The bitmap 01000001 can for example stand for the binary-coded number 65, for the BIT-coded number 41, for the ASCII character A, for two keys pressed simultaneously or for the request to turn on the buzzer.

Handshaking

The header consists of the following 16 bits:



The first header byte contains the handshake information for both applications.
The second header byte contains the index and describes the contents of the subsequent data part.

The following description of the bits of byte 1 refers to both the PLC and the KeTop T40. Therefore device may be the PLC or the KeTop T40 depending on the direction.

On-line (bit 7)

This bit indicates that the device is ready for receiving. If the device is not on-line, no messages should be sent to the device. Irrespective of that, the device may send data. The on-line bit and the fault bit do not depend on each other.

Fault (bit 6)

This bit indicates a fault on the device. If the device is on-line a fault code can be requested.

Handshake receiving (bit 5)

This bit is set after a message is received. With each message received, the state of this bit changes:
When the first message is received, the state changes from 0 to 1. When the next message is received, the state changes from 1 to 0 again.

Handshake sending (bit 4)

This bit signalizes the sending of a new message. With each message sent, the status of this bit changes:
When the first message is sent, the state changes from 0 to 1. When the next message is sent, the state changes from 1 to 0 again.

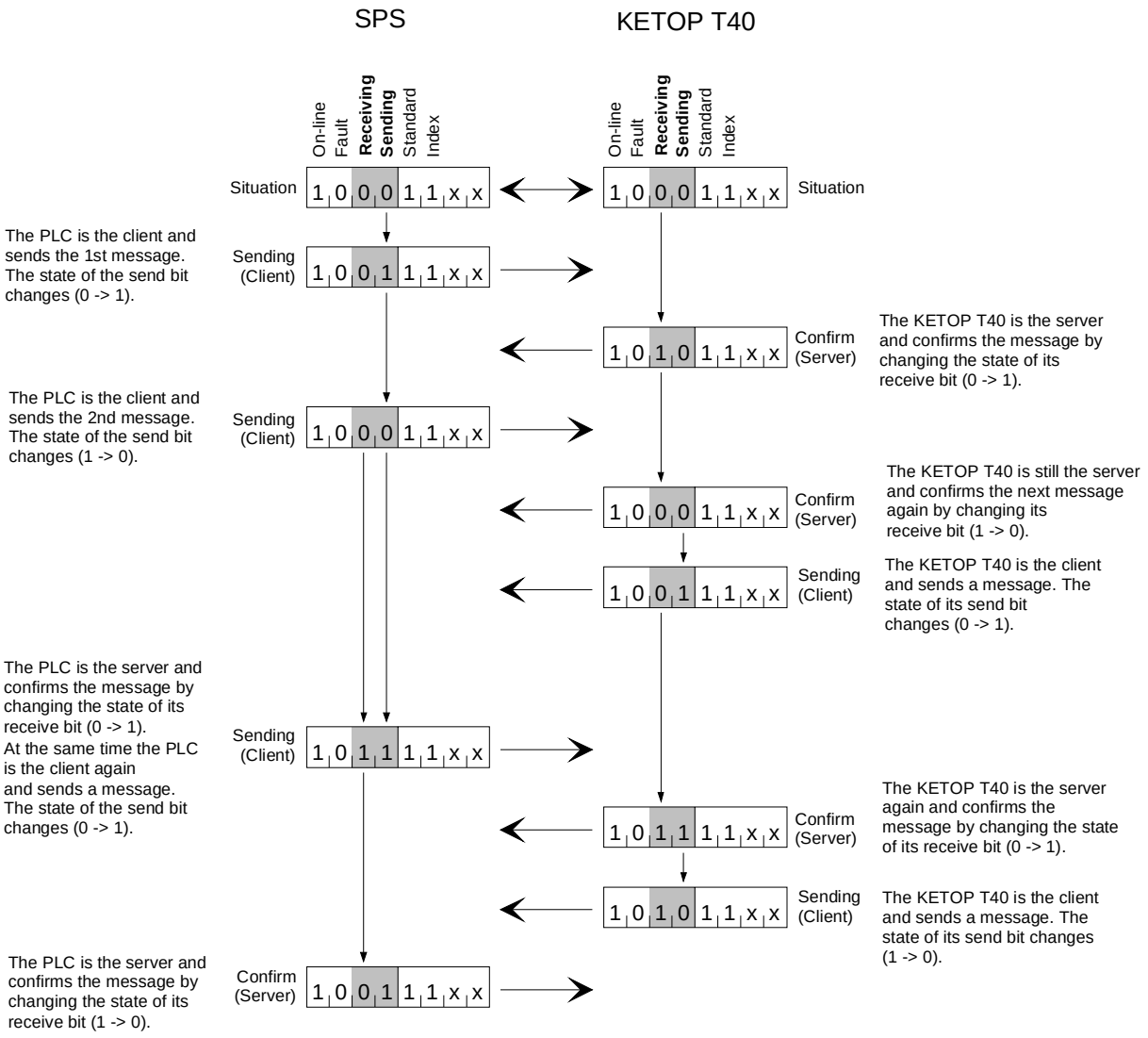
Standard bit (bit 3)

This bit defines the assignment of the data field for the process data transmission. In this case, the significance of the PD index is standardized. If this bit is not set, that means that the significance of the field is manufacturer-specific.

Index bit (bit 2)

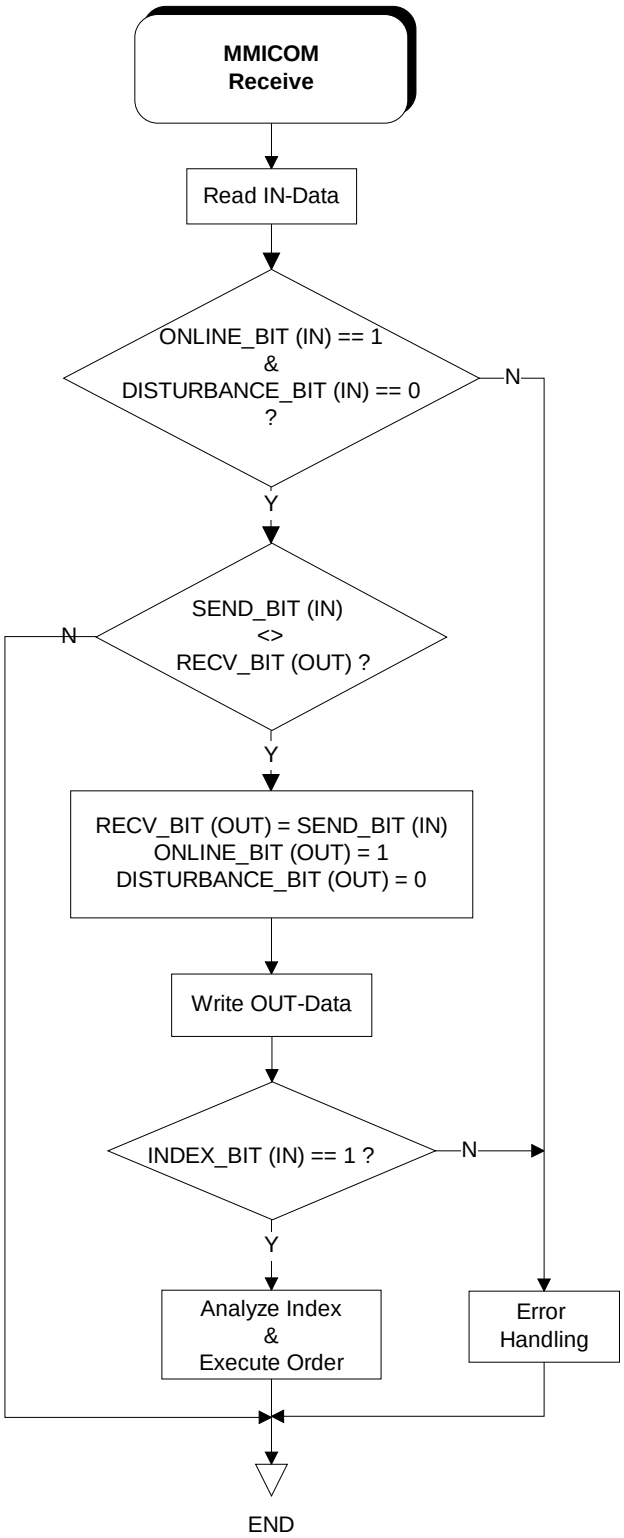
This bit defines the existence of an index field in the byte 2 of the header byte. Therefore it is always set to 1.

The following diagram shows an example of handshaking in the MMI-COM protocol between PLC and KeTop T40.



Handshaking between PLC and KeTop T40

a) Receiving Data



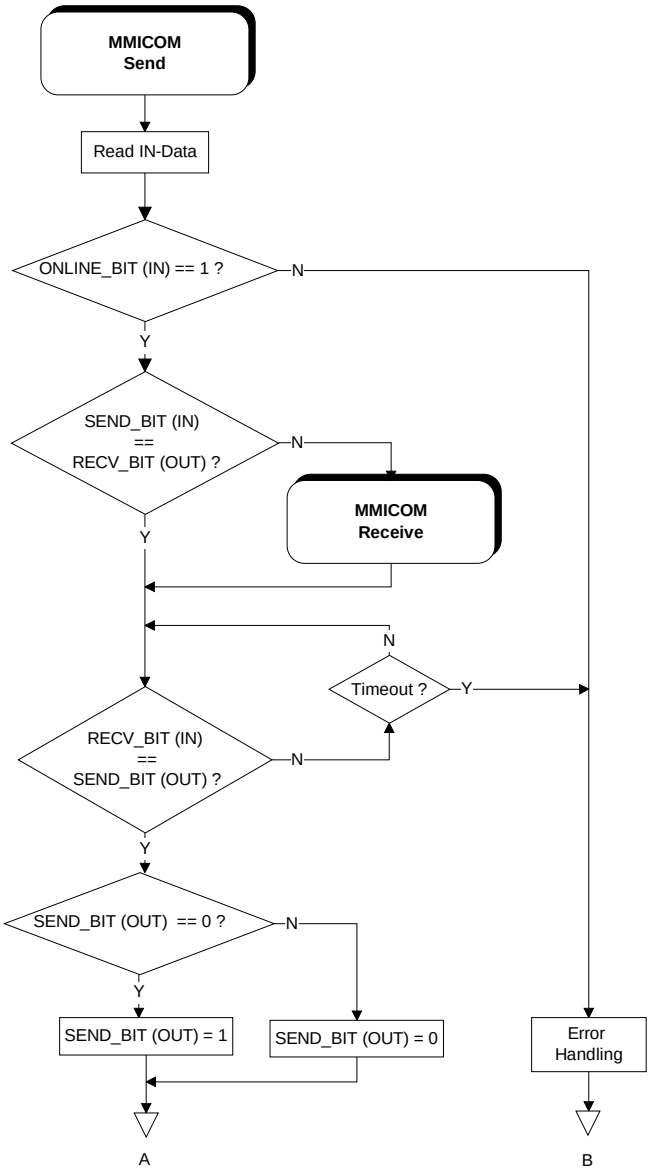
Handshaking when receiving data

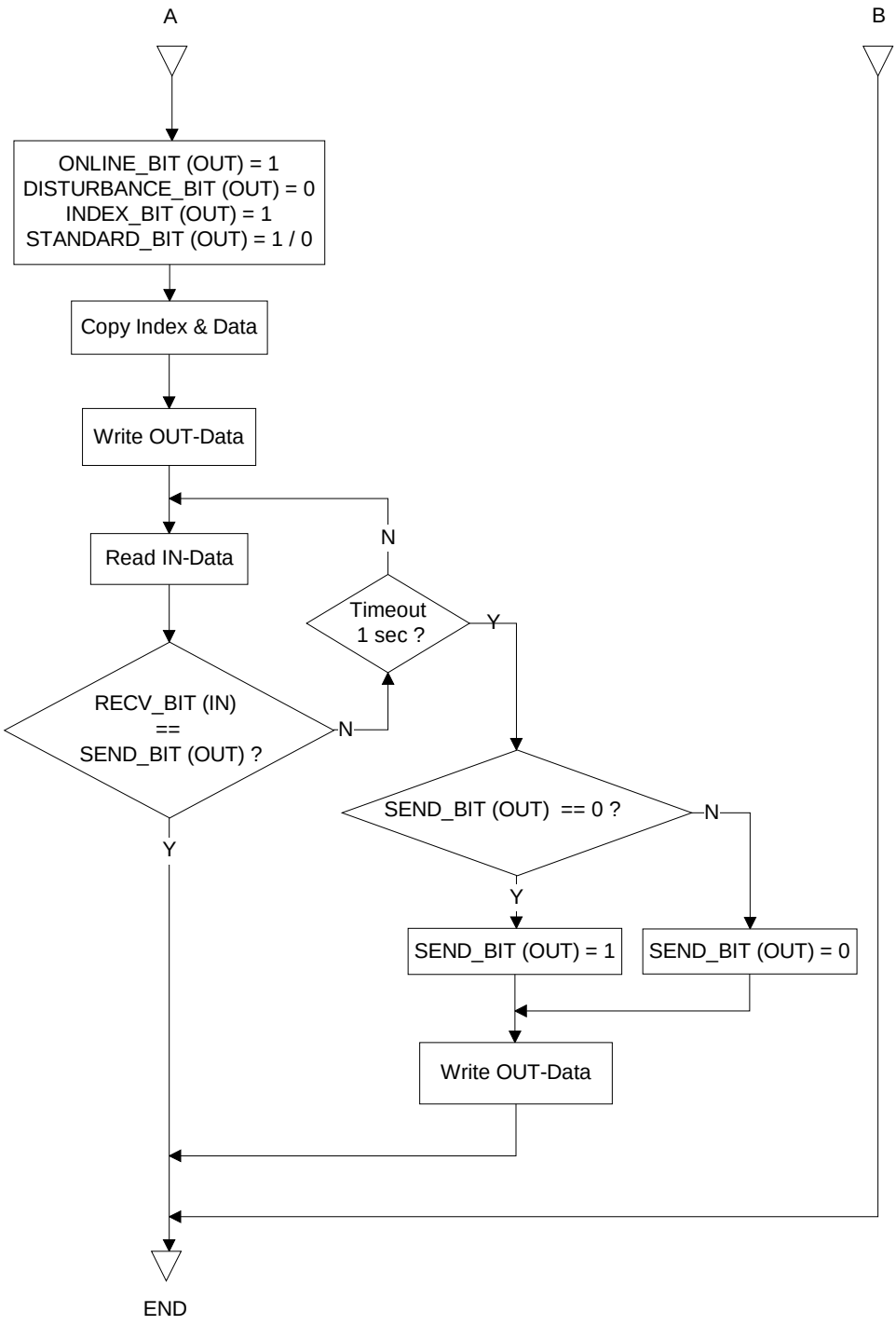
b) Sending Data

The KeTop T40 tests if the receive bit of the partner station has the same state as the own send bit. If this is the case the device will be able to send data. The corresponding data and the index will be copied to the output buffer.

Notice
To avoid inconsistent data in the message at bus systems with cyclic data transfer (e.g. PROFIBUS, InterBus), the first header byte containing the toggled send bit must always be written as last byte.

Following that the send bit of the send station will be inverted to indicate the sending. The handshake of the partner statioin is monitored. If the counterpart station does not respond within 1 s, the send bit will be toggled.


MMICOM
Send
Read IN-Data
ONLINE_BIT (IN) == 1 ?
N
Y
SEND_BIT (IN)
==
RECV_BIT (OUT) ?
N
Y
MMICOM
Receive
N
Timeout ?
Y
RECV_BIT (IN)
==
SEND_BIT (OUT) ?
N
Y
SEND_BIT (OUT) == 0 ?
N
Y
SEND_BIT (OUT) = 1
SEND_BIT (OUT) = 0
Error
Handling
A
B



Handshaking when sending data

Overview of Functions

The following table shows an overview of the services that constitute the base of the particular handheld terminal functions.
Services are request telegrams that are sent from a send station (client) to a receive station (server). In the data fields, the user data defined by the index are transmitted.

Standard Objects

Header Byte 1 Reserved	Byte 2 Index (HEX)	Data fields Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Keypad							
PLC <- HT	10: Send key status	Key status (group 1) keys 1 to 8	Key status (group 2) keys 9 to 16	Key status (group 3) keys 17 to 24	Key status (group 4) keys 25 to 32		
PLC <- HT	12:Send key number	Length=1	Key number				
PLC <- HT	13: Send key code	Length=2		Key code			
LEDs							
PLC -> HT	20: Send LED status	LED status (group 1)	LED status (group 2)	LED status (group 3)	LED status (group 4)	LED status (group 5)	
PLC -> HT	21:Send LED group	Number of groups n	LED group No. x	LED status group x	LED status group x+1	...	LED status group x+ (n/8)
PLC -> HT	22: Set LED color		LED attribute				
PLC -> HT	23: Send LED No.	LED attribute	LED No.				
Texts							
PLC -> HT	25: Display text No.		Display attribute	Text number (binary-coded)			

Header		Data fields					
Byte 1 Reserved	Byte 2 Index (HEX)	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Variables							
PLC -> HT	14: Send 1-byte variable	b15 b0 Variable No.		b7 b0 Value of var.			
PLC -> HT	15: Send 2-byte variable	b15 b0 Variable No.		b15 b0 Value of variable			
PLC -> HT	16: Send 4-byte variable	b15 b0 Variable No.		b31 b0 Value of variable			
PLC -> HT	17: Send floating point	b15 b0 Variable No.		1st byte	2nd byte	3rd byte	4th byte
PLC -> HT	18: Send bytes	Length	Variable No.	1st byte	2nd byte	...	...
PLC -> HT	2A: Display 1-byte variable	b15 b0 Variable No.		b7 b0 Value of var.			
PLC -> HT	2B: Display 2-byte variable	b15 b0 Variable No.		b15 b0 Value of variable			
PLC -> HT	2C: Display 4-byte variable	b15 b0 Variable No.		b31 b0 Value of variable			
PLC -> HT	2D: Display floating point	b15 b0 Variable No.		1st byte	2nd byte	3rd byte	4th byte
PLC -> HT	2E: Display bytes	Length	Variable No.	1st byte	2nd byte	...	...
PLC <-> HT	40: Request 1-byte variable	b15 b0 Variable No.					
PLC <-> HT	41: Request 2-byte variable	b15 b0 Variable No.					
PLC <-> HT	42: Request 4-byte variable	b15 b0 Variable No.					
PLC <-> HT	43: Request floating point	b15 b0 Variable No.					
PLC <-> HT	44: Request bytes	Length	Variable No.				
Text monitor							
PLC -> HT	28:Send text characters	Number of characters	Display attribute	Text characters	...	...	...
PLC -> HT	29: Position cursor	Cursor x-position	Cursor y-position				

Manufacturer-Specific Objects

Header		Data fields					
Byte 1 Reserved	Byte 2 Index (HEX)	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Texts							
PLC -> HT	A0: Display text number	Line	Display attribute	Text number binary		Number of follow-up texts	0
Graphic functions							
PLC -> HT	B0: Pixel	x	y				
PLC -> HT	B1: Line	x0	y0	x1	y1		
PLC -> HT	B2: Rectangle	x0	y0	x1	y1	fill	
PLC -> HT	B3: Circle	x	y	r			
PLC -> HT	B4: Ellipsis	x	y	rx	ry	fill	
PLC -> HT	B5: Text 1	x	y	len	Font size	Font attribute	
PLC -> HT	B6: Text 2	Text character	....	....	....	....	...
PLC -> HT	BA: Set line type	Bitmap of line					
PLC -> HT	B9: Set color	fg	bg				
PLC -> HT	B8: Clear window	x0	y0	x1	y1		
SPS -> HT	BB: Display Bitmap	Bitmap-No.		x	y		
General functions							
PLC -> HT	A4: Clear screen						

Serial Side (KeTop -> Gateway)

Structure of Telegrams

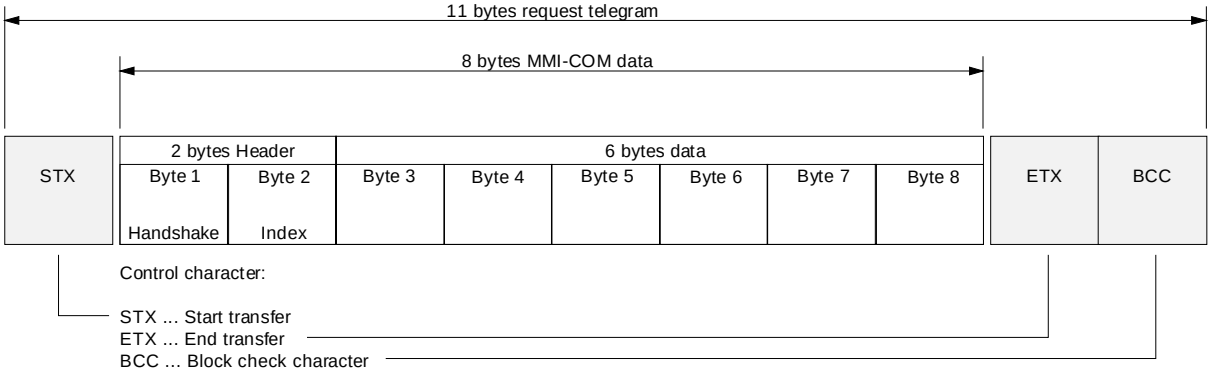
Request Telegram

The request telegrams always have a length of 11 bytes. They consist of:

- **3 bytes of control characters** (STX, ETX, BCC) and
- **8 bytes of MMI-COM data** (2 bytes header und 6 bytes data part)

The first byte contains the handshake information between the handheld terminal and the PLC.

The second byte in the header contains the index number which stands for a certain command. The structure of the data fields (user data) is defined by the corresponding index.



The description of the request telegram in the chapters "HT System Variables“ and "HT Functions" only refers to the MMI-COM data part without control characters.

Confirmation Telegram

After receipt of a message, the BCC wil be checked, and, depending on the result, a positive or negative acknowledgement will be sent.

ACK	positive confirmation (BCC correct)
NAK	negative confirmation (BCC wrong)

If the confirmation telegram is negative (NAK) the request telegram must be repeated.

Checksum Calculation

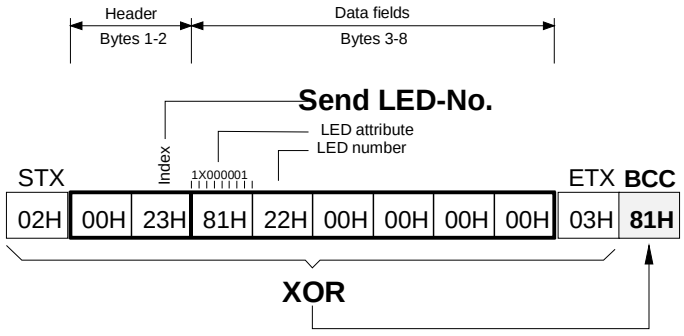
The value of the BCC byte is calculated by an XOR logic of the telegram bytes from STX up to and including ETX.

Example

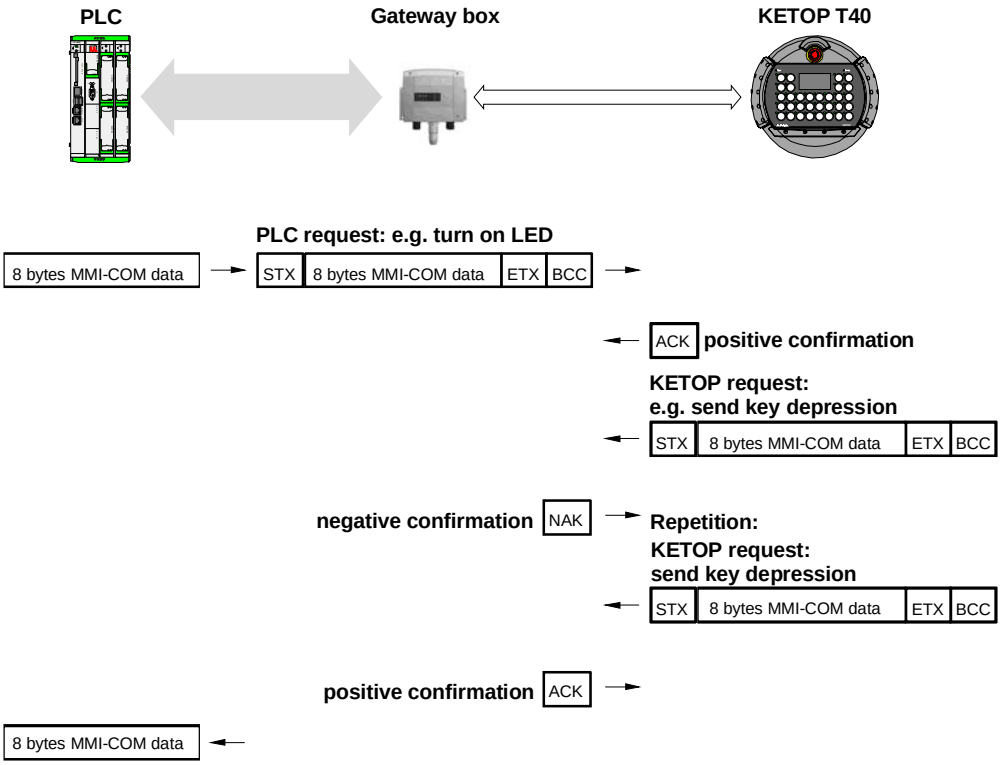
The red LED No. 34 on the handheld terminal should be turned on by using the service 23H "Send LED No.".

	Index	Data fields					
PLC -> HT	23: Send LED No. 23H	LED attribute xx	LED No xx				

The service 23H "Send LED No." contains the following HEX values:



Example: Serial Protocol



Example: serial protocol

Bus Side (Gateway <-> SPS)

The data transport between gateway and PLC depends on the bus system used.

Notice
At bus systems with cyclic data transfer (PROFIBUS-DP, InterBus) the consistency of the data to be sent must be ensured on the PLC side. That means, the send bit in the MMI-COM header must only be toggled when the complete subsequent data part is contained in the output buffer.

6 HT System Variables

System variables offer additional functions to the user. If the system variables are sent with the corresponding index, they can activate and deactivate certain functions on the handheld terminal (e.g. automatic sending of keycodes).

The system variables range from 1 to 99 and must not be used by the user.

System variables distinguish 1-byte, 2-byte and 4-byte variables.

Var. No.	Designation	Data type	Access HT	Access PLC	Default value	Description
1-byte variables						
1 (01H)	reserved	-	-	-	-	-
2 (02H)	reserved	-	-	-	-	-
3 (03H)	reserved	-	-	-	-	-
4 (04H)	reserved	-	-	-	-	-
5 (05H)	reserved	-	-	-	-	-
6 (06H)	BEEP	BOOL		R/W	FALSE	Buzzer activation (permanent tone)
7 (07H)	KEY_CLICK_PRESSED	BOOL		R/W	FALSE	Keyclick when a key is pressed
8 (08H)	SEND_PRESSED_KEY_NR	BOOL		R/W	FALSE	The HT sends the key number when a key is pressed.
9 (09H)	SEND_RELEASED_KEY_NR	BOOL		R/W	FALSE	The HT sends the key number when a key is released.
10 (0AH)	reserved	-	-	-	-	-
11 (0BH)	reserved	-	-	-	-	-
12 (0CH)	reserved	-	-	-	-	-
13 (0DH)	SEND_KBD_MAP	BOOL		R/W	FALSE	The HT sends the key bitmap when a key is pressed or released.
14 (0EH)	reserved	-	-	-	-	-
15 (0FH)	reserved	-	-	-	-	-
16 (10H)	reserved	-	-	-	-	-
17 (11H)	SYSTEM_RESET	UINT8		W	-	System reset at bitmap 55H
18 (12H)	FETCH_TEXT_VAR	BOOL		W	FALSE	In case of a text call, the HT requests once all output variables contained in the text.
19 (13H)	SEND_PRESSED_KEY_CODE	BOOL		R/W	FALSE	The HT sends the key code when a key is pressed.
20 (14H)	ORDER_BOX	UINT8		W	-	Order box of master station: 1 ... Request key bitmap
21 (15H)	KEY_CLICK_RELEASED	BOOL		R/W	FALSE	Keyclick when a key is released.
22 (16H)	EVENT_VAR_NOT_ON_SCREEN	BOOL		R/W	FALSE	Event message indicating when a variable is written, for which no output field is provided on the display => EVENT_CODE

Var. No.	Designation	Data type	Access HT	Access PLC	Default value	Description
23 (17H)	DISABLE_MENU	BOOL		R/W	FALSE	Enabling (0) or disabling (1) the activation of the main menu (activation by pressing 1st and 4th key)
24 (18H)	DISABLE_EDITOR	BOOL		R/W	FALSE	Enabling(0) or disabling(1) of variables input on HT
25 (19H)	reserved	-	-	-	-	-
26 (1AH)	reserved	-	-	-	-	-
27 (1BH)	reserved	-	-	-	-	-
28 (1CH)	EVENT_MENU	BOOL		R/W	FALSE	Event message on menu access and exit => EVENT_CODE
29 (1DH)	KBD_LOCKED	BOOL		R/W	FALSE	Enabling (0) or disabling (1) of the keypad on the HT
30 (1EH)	NEXT_FIELD_AFTER_ENTER	BOOL		R/W	TRUE	Cursor appears in next input field after pressing Enter.
31 (1FH)	SEND_RELEASED_KEY_CODE	BOOL		R/W	FALSE	The HT sends the key code when a key is released.
32 (20H)	AUTO_KEY_REPEAT	BOOL		R/W	FALSE	Automatic repetition while key is pressed.
2-byte variables						
1 (01H)	reserved	-	-	-	-	-
2 (02H)	FETCH_TEXT_VAR_CYCLE_TIME	UINT16		R/W	0	Cycle time in ms at which the HT requests all output variables displayed.
3 (03H)	BEEP_TIME	UINT16		R/W	0	Buzzer activation (duration of beep in ms)
4 (04H)	APPLICATION_SW_VERSION	UINT16		R	-	Version of HT application software Vx.y: High Byte -> x, Low Byte -> y
5 (05H)	DEVICE_TYPE	UINT16		R	-	40 ... KeTop T40
6 (06H)	reserved	-	-	-	-	-
7 (07H)	reserved	-	-	-	-	-
8 (08H)	AUTO_KEY_DELAY_TIME	UINT16		R/W	250	Repetition rate in ms.
4-byte variables						
1 (01H)	ERROR_CODE	UINT32	W	R		Error message HT -> PLC (not used at present)
2 (02H)	EVENT_CODE	UNIT32	W	R		Event message HT -> PLC: Bit 24 - 31: event number Bit 0 – 23: event-related information

Access

R	Read
W	Write

Value ranges of data types used

BOOL	Boolean Number (0 = FALSE, 1 = TRUE)
UINT8	integral number without sign (0...255)
UINT16	integral number without sign (0...65535)
UINT32	integral number without sign (0...4294967295)

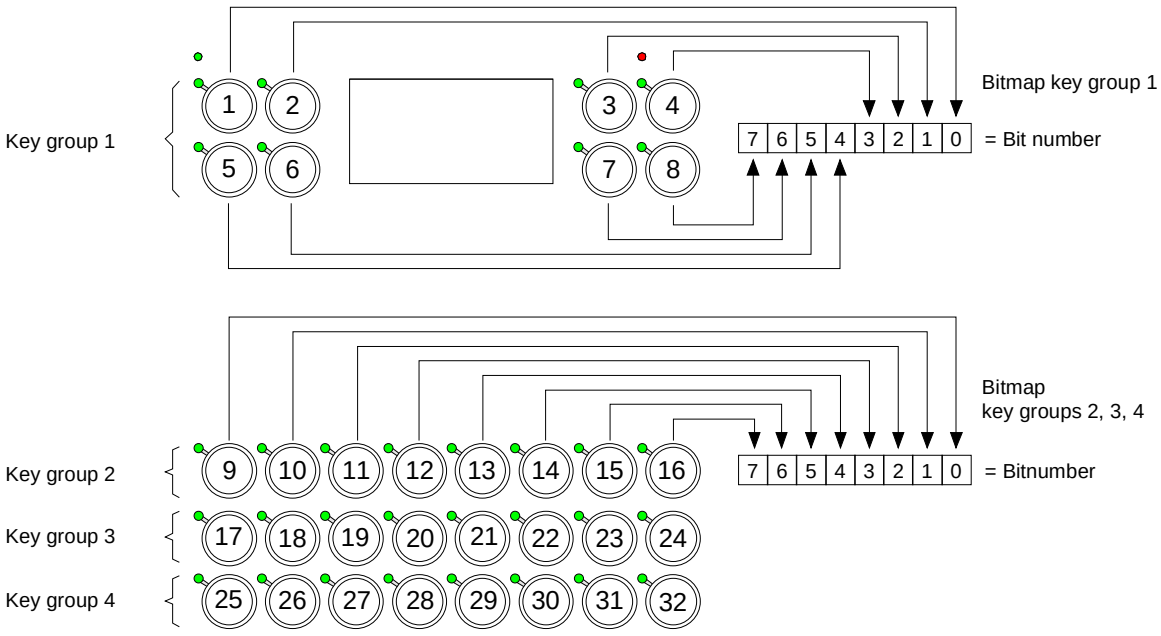
7 HT Functions

The request telegrams always have a length of 11 bytes (2 bytes header + 6 bytes data fields=8 bytes MMI-COM data and 3 bytes control characters). The description of the request telegrams in the following chapters only refers to the 8-byte MMI-COM data.

Keypad

The following drawing shows the physical grouping of the HT keys and the assignment of the particular keys to the bits of the parameter "key group n". At the KeTop T40 this grouping and assignment is identical for the LEDs of the keypad.

Physical Grouping



Key groups of KeTop T40 keypad

Audible Keyclick

When the 1-byte system variable No. 7 (07H) "KEY_CLICK_PRESSED" is set, the handheld terminal confirms the **pressing** of a key with an audible signal. When the system variable No. 21 (15H) "KEY_CLICK_RELEASED" is set, the HT confirms the **releasing** of a key with an audible signal.

The telegram required for setting the system variable is structured as follows:

for KEY_CLICK_PRESSED:

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	07H	b7 b0 Value of var. 01H			

for KEY_CLICK_RELEASED:

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No 00H	15H	b7 b0 Value of var. 01H			

For deactivating the keyclick, the corresponding system variable must be reset. The telegram is sent once again, but this time with the variable value 00H.

Variable No. This parameter identifies the variable.
Value of variable This parameter contains the value of the variable.

Sending Key Status to the PLC

When the 1-byte system variable No.14 (0DH) "SEND_KBD_MAP" is set on the HT, the key bitmap is automatically sent to the PLC when a key is pressed or released on the HT.

The telegram required for setting the system variable is structured as follows:

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	0DH	b7 b0 Value of var. 01H			

From that time on, a telegram with the current key status will be sent to the PLC each time a key is pressed or released on the handheld terminal (service 10H: "Send key status"):

	Index	Data fields					
PLC <- HT	10: Send key status 10H	Key status (group 1) xx	Key status (group 2) xx	Key status (group 3) xx	Key status (group 4) xx		

Key status (group n) This parameter contains the key status of the corresponding key group. For the assignment of the keys, see drawing in chapter „Keypad“.

Requesting Key Status from the HT

When the value 1 is written to the 1-byte system variable No. 20 (14H) "ORDER_BOX" on the HT, the HT sends the current key bitmap once to the PLC.

The telegram required for requesting the HT key status is structured as follows:

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	14H	b7 b0 Value of var. 01H			

After receiving the telegram, the HT responds once using the service 10H "Send key status":

	Index	Data fields					
PLC <- HT	10: Send key status 10H	Key status (group 1) keys 1-8 xx	Key status (group 2) keys 9-16 xx	Key status (group 3) keys 17-24 xx	Key status (group 4) keys 25-32 xx		

Sending Key Number to the PLC

2 possibilities are available for sending a key number to the PLC:

When the 1-byte system variable No. 8 (08H) "SEND_PRESSED_KEY_NR" is set on the HT, the HT sends the key number to the PLC each time the corresponding key is **pressed** on the HT. The variable must be set once only (e.g. in the power-up phase).

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	08H	b7 b0 Value of var. 01H			

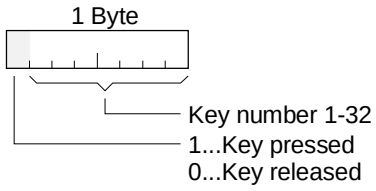
When the 1-byte system variable No. 9 (09H) "SEND_RELEASED_KEY_NR" is set on the HT, the HT sends the key number each time the corresponding key is **released** on the HT. The variable must be set once only (e.g. in the power-up phase).

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	09H	b7 b0 Value of var. 01H			

After receiving one of the two telegrams, the HT sends, each time a key is pressed or released, the key number (1-32) to the PLC using the service 12H "Send key number":

	Index	Data fields					
PLC <- HT	12:Send key number 12H	Length=1 01H	Key number xx				

Key number This parameter contains the corresponding physical key number (see drawing in chapter "Keypad") when the status of a key changes (key is pressed or released). The key number ranges from 1 to 32. The highest-order bit contains the information indicating whether the key has been pressed or released:



Sending Key Code to the PLC

When the 1-byte system variable No. 19 (13H) "SEND_PRESSED_KEY_CODE" is set on the HT, the HT automatically sends the logic key code defined during programming to the PLC each time

a key is **pressed** on the HT. For sending, the service 13H "Send key code" is used. The variable must be set once only (e.g. in the power-up phase).

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	13H	b7 b0 Value of var. 01H			

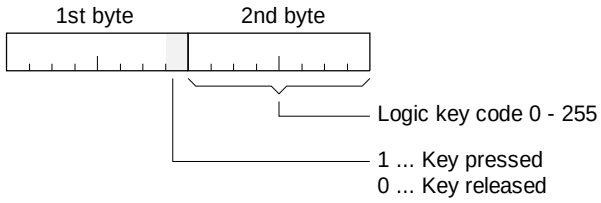
When the 1-byte system variable No. 31 (1FH) "SEND_RELEASED_KEY_CODE" is set on the HT, the HT sends the key number each time the corresponding key is **released** on the HT. The variable must be set once only (e.g. in the power-up phase).

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	1FH	b7 b0 Value of var. 01H			

After receiving one of the two telegrams, the HT sends, each time a key is pressed or released, the key number (1-32) to the PLC using the service 13H "Send key code":

	Index	Data fields					
PLC <- HT	13: Tasten-code senden 13H	Länge=2 02H		Tastencode xx	xx		

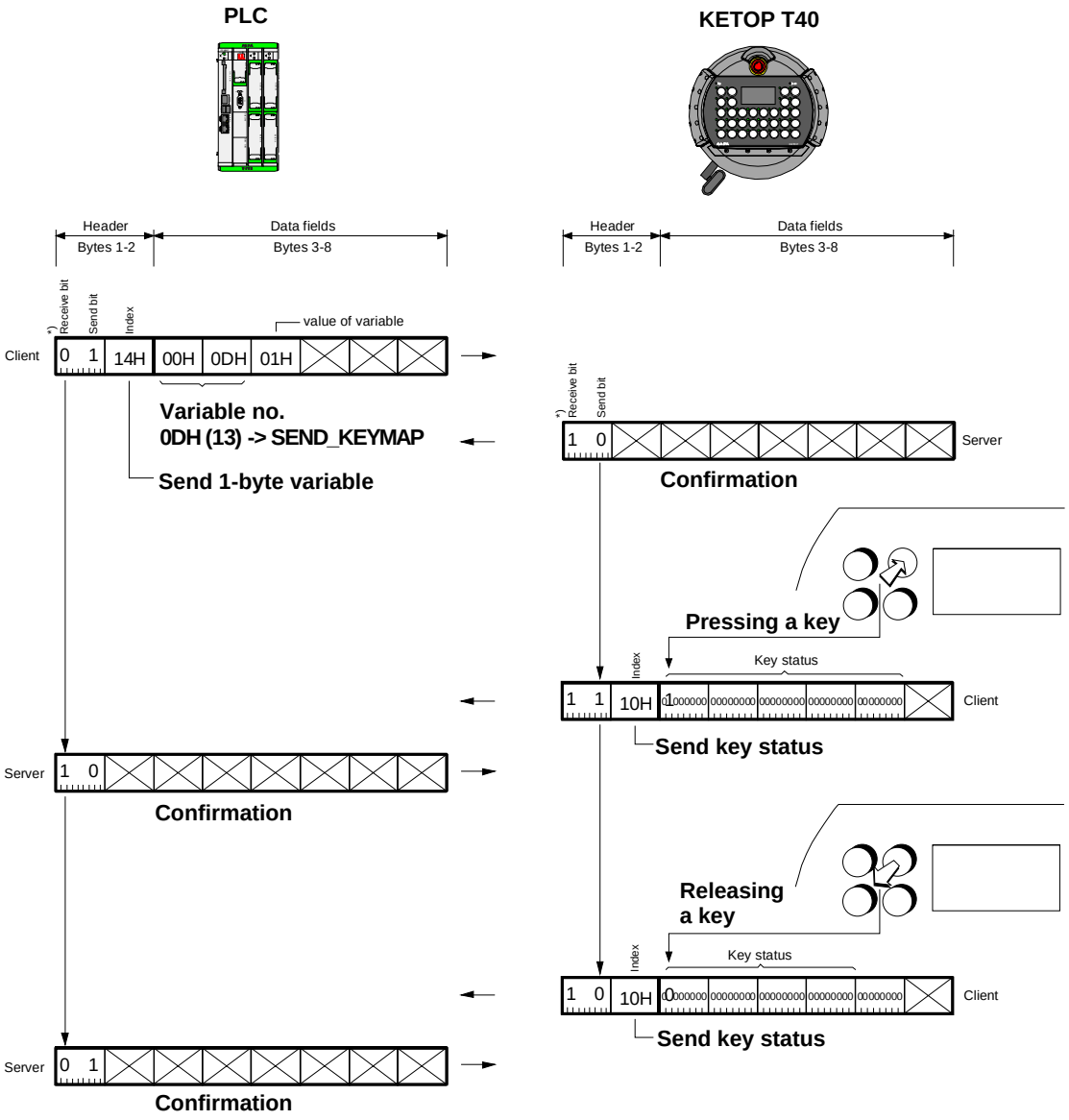
Key code When the key status changes (key is pressed or released), this parameter contains the logic key code (according to the configured keypad layout).



Example: Sending Key Status of HT to the PLC

The handheld terminal should send the entire key bitmap to the PLC each time a key is pressed and released.

This function is for example suitable for a jogging mode to be executed via the handheld terminal. The pressing and releasing of the key is transmitted to the process data image (memory) of the PLC.



**) To get a better overview, only the receive bit and the send bit of the 1st byte of the control/status word are shown.*

Example: Sending key status of the KeTop T40 to the PLC

After turning on, the HT does NOT send the key bitmap automatically when the key status changes. For activating this function, the PLC must set the 1-byte control variable with number 13 in the KeTop T40 to 1 (=TRUE).

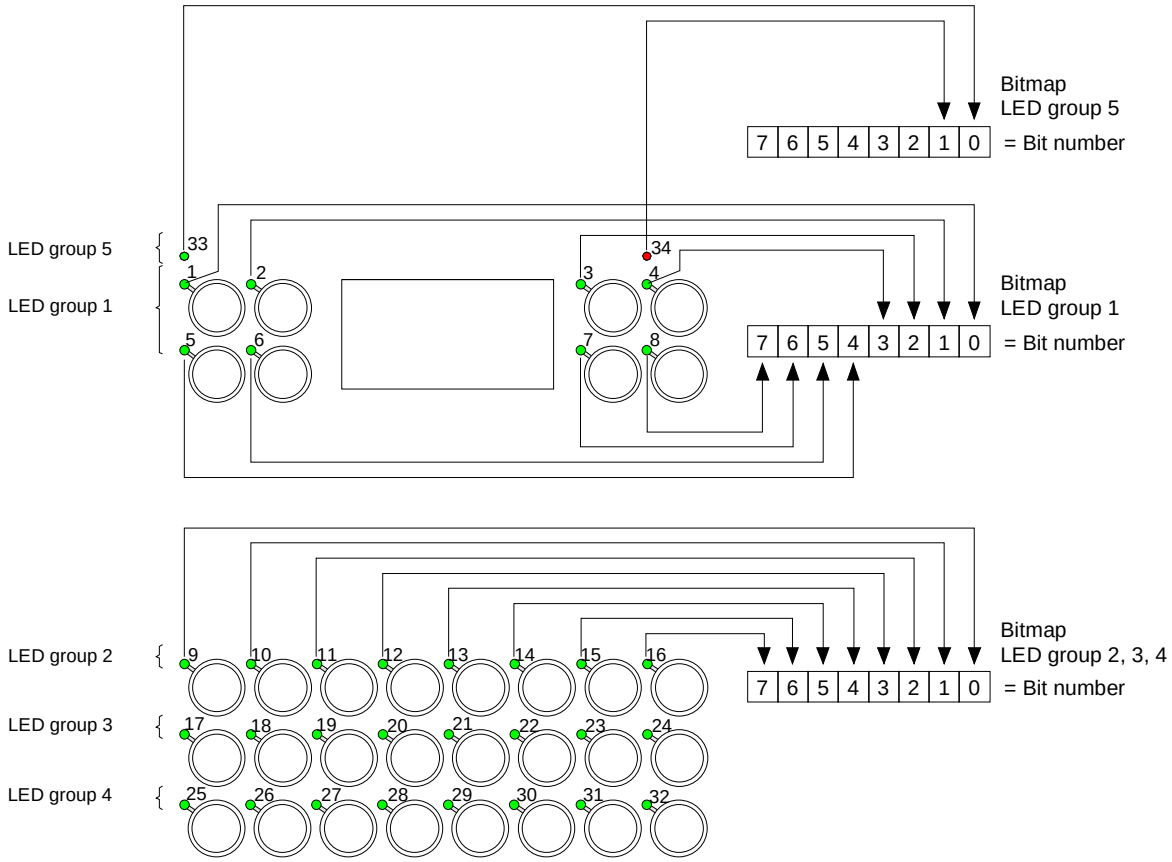
The handheld terminal confirms the receipt of the telegram by sending back a telegram to the PLC. From that time on, the function "SEND KEY MAP" is activated.

When a key is pressed now, the handheld terminal sends the current key bitmap to the PLC which acknowledges the receipt. When the key is released, the handheld terminal sends the key bitmap again, and the PLC acknowledges the receipt again.

LEDs

The following drawing shows the physical grouping of the LEDs on the KeTop T40 keypad and the assignment of the particular LEDs to the bits of the parameters "LED group n".

Physical Grouping



LED groups of KeTop T40

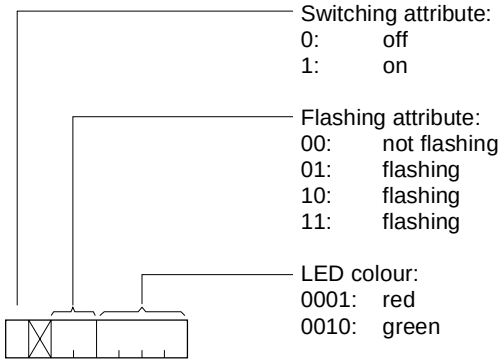
Notice
The LEDs of the keys „Shift Lock“, „Ctrl Lock“ and „Alt Lock“ are activated internally and cannot be activated by the PLC.

Activating Particular LEDs on the HT

For activating LEDs on the handheld terminal, the service 23H is used. Using this service, the PLC sends the number of the LED to be activated and the LED attribute to the handheld terminal.

	Index	Data fields					
PLC -> HT	23: Send LED No. 23H	LED attribute xx	LED No. xx				

LED attribute This parameter sets the switching attribute, the flashing attribute and the LED color. This LED attribute is only valid for the service 23H.

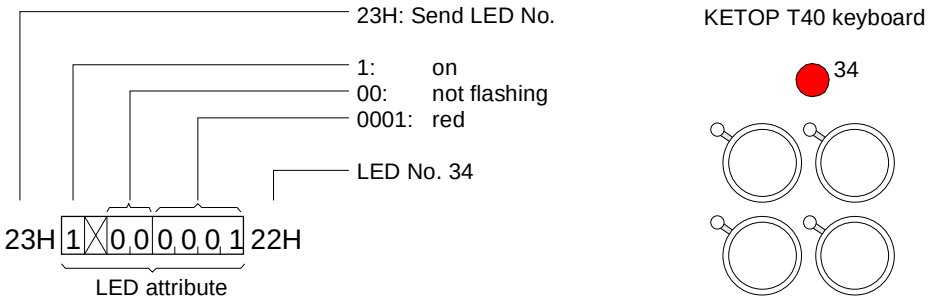


LED No. This parameter contains the number of the LED to be activated.

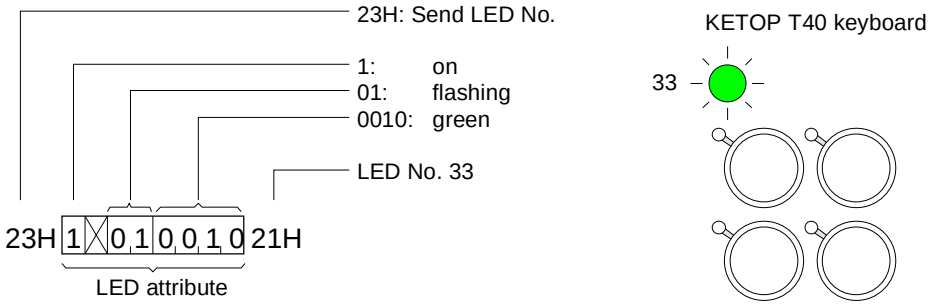
Example: Defining LED Attribute in Service 23H

Using the service 23H, the red LED Number 34 should be activated first and then the green LED number 33 switched over to the flashing mode. The corresponding LED attributes must be defined as follows:

LED attribute for activating the red LED No. 34:



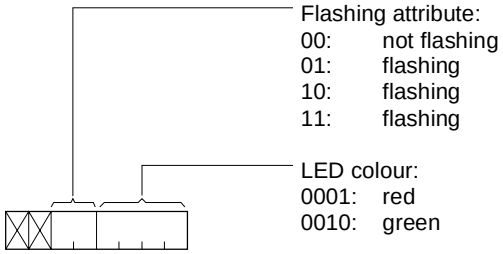
LED attribute for flashing mode of green LED No. 33



Activating LED Groups and All LEDs on the HT

The LEDs can be activated group by group or all at once. In both cases, the LED attribute must be defined before.

LED-Attribut The LED attribute differs to that of service 23H (see previous chapter). Only the flashing attribute and the LED color are defined. The switching attribute cannot be set here as it is contained in the parameters "LED groups No.x" (service 21H) and "LED status group x" (service 20H).



The LED attribute defines the status of the selected LEDs and what status of the active LEDs should be deactivated again.

Notice
The attribute "flashing" has the priority over "not flashing".

When the LEDs are switched off, the LED attribute must contain the actual LED status (lighting or flashing).

The service 20H "Send LED status" behaves like the service 21H "Send LED groups" whereby the service 20H always sends all LED groups.

a) Activating LED Groups on the HT

For activating LED groups on the handheld terminal, the service 21H "Send LED group" is used. Before a LED group is activated, the LED attribute must be defined and sent to the HT using the service 22H "Set LED color":

	Index	Data fields					
PLC -> HT	22: Set LED color 22H		LED attribute xx				

Color and flashing attribute of the LED group are set now, and therefore the LED group is ready for being activated.

Notice
The parameter LED color is relevant for LED No. 34 only, because this LED is the only 2-color LED.

	Index	Data fields					
PLC -> HT	21:Send LED group 21H	Number of LEDs n xx	LED group No. x xx	LED group x xx	LED group x+1 xx	... ...	LED group x+ (n/8) xx

Number of LEDs n This parameter contains the number of LEDs (n=8, 16, 24 or 32) to be activated (max. 4 groups).

LED group No. x Specifies the first LED group No. to be activated.

LED group x This parameter contains the status of the corresponding LED group. The assignment of the particular LEDs to the bits of this parameter is shown in the drawing one page before.

b) Activating All LEDs on the HT

For activating all LEDs on the handheld terminal, the service 20H is used. This service contains the LED statuses of all LED groups. Before all LEDs can be activated, the LED attribute must be defined and sent to the HT using the service 22H "Set LED color":

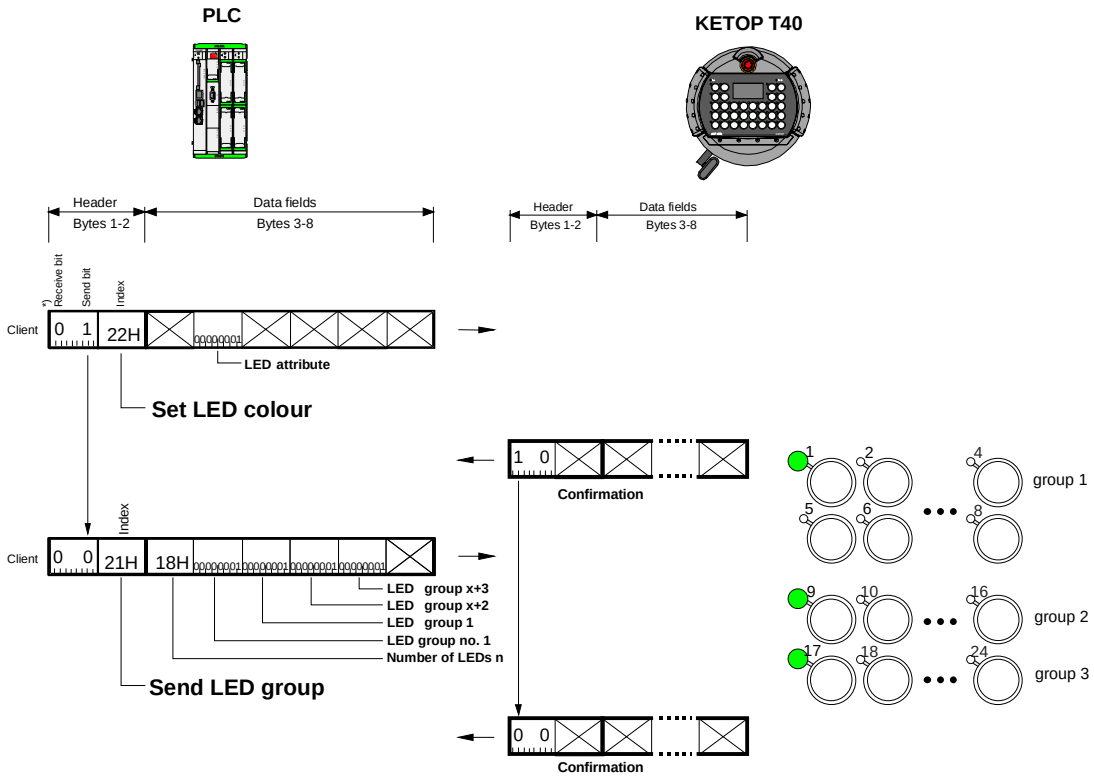
	Index	Data fields					
PLC -> HT	22: Set LED color 22H		LED attribute xx				

The color and the flashing attribute of all LEDs are set now, and therefore the LEDs are ready for being activated.

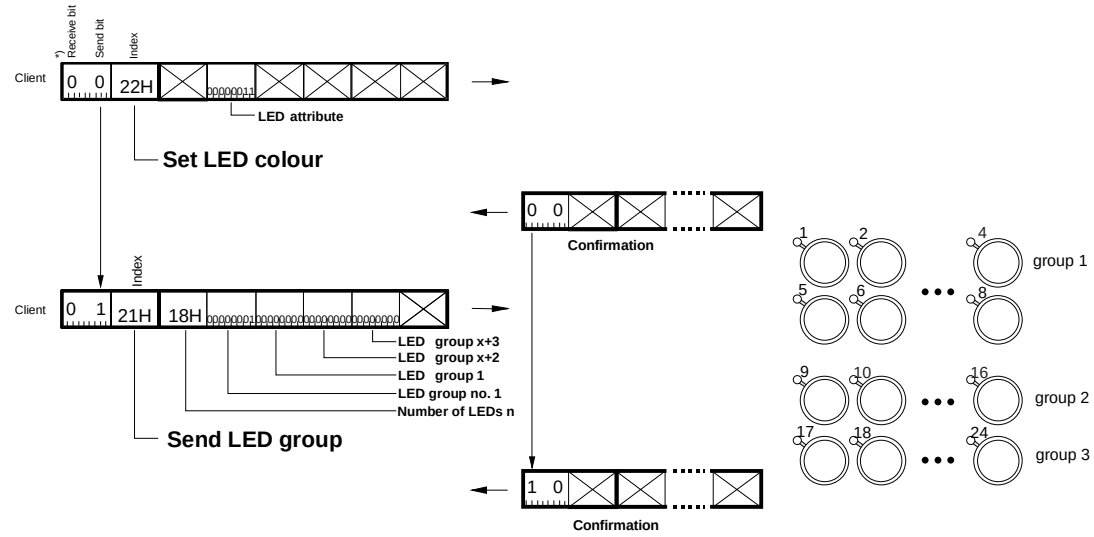
	Index	Data fields					
PLC -> HT	20: Send LED status 20H	LED status (group 1) xx	LED status (group 2) xx	LED status (group 3) xx	LED status (group 4) xx	LED status (group 5) xx	

Example: Activating LED Groups Using the Service 21H

a) The first LEDs of the groups 1 to 3 should light:



b) All LEDs lit should be turned off.



**) To get a better overview, only the receive bit and the send bit of the 1st byte of the control/status word are shown.*

Texts

Texts are created on a PC using the programming software and loaded into the handheld terminal. The PLC can call texts line by line by sending text and line number to the HT where these numbers will be displayed.

Furthermore the handheld terminal offers the possibility to display and alter variables directly on the device. In the programming software, input and output fields for the variables can be defined in the text lines.

The KeTop T40 provides 192kByte memory space for variables and texts.

Memory requirements

text	24 bytes
variable	6 bytes

e.g.: when 5000 texts are created, memory space for about 12500 variables is left.

Text Variables

The numbers of text variables range from 100 to 65535 (for system - variables, the range from 1 to 99 is reserved). The following variables are distinguished:

- Input variable (HT -> PLC)
- Output variable (PLC -> HT)
- Input/output variable (PLC <-> HT)

Displaying texts on the KeTop T40

a) Standard object

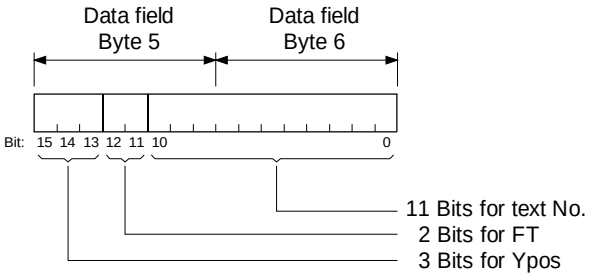
For displaying a text on the handheld terminal, the service 25H is used. This telegram contains the text number under which the text to be displayed is stored in the text memory of the HT.

	Index	Data fields				
PLC -> HT	25: Display text No. 25H		Display attribute 01H	Text number (binary-coded) xx | xx		

After receiving this telegram, the handheld teminal displays the corresponding text.

Display attribute
For displaying a text, this parameter must be set to 01H.

Text number (binary-coded)
Besides the text number, this parameter specifies the line in which the text should be displayed and the text length (eventually several lines). The binary coding is defined as follows:



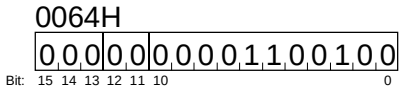
Text No.
Actual text number. The maximum value depends on the number of defined variables, because they are stored in the same memory space.

FT
Number of follow-up texts (text numbers in ascending order).

Ypos
Selection of display line in which the text should be displayed (0 -> 1st line, 1 -> 2nd line, etc.).

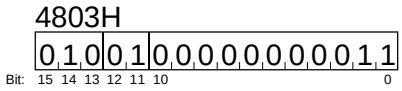
Example 1

Text number 0064H -> Text No.=100, FT=0, Ypos=0.
The text with number 100 is displayed in the 1st display line.



Example 2

Text number 4803H -> Text No.=3, FT=1, Ypos=2. ==> In this case the text with number 3 is displayed in the 3rd display line and the text with number 4 in the 4th line.



b) Manufacturer-Specific Object

If a text display on the handheld terminal is activated with the manufacturer-specific service A0H the text number must be specified directly. The line and the quantity of follow-up texts are specified in a byte especially provided for that purpose.

	Index	Data fields					
PLC <- HT	A0: Display text number	Line	Display attribute	Text number binary		Quantity of follow-up texts	
	A0H	xx	xx	xx	xx	xx	0

Displaying a Text with Output Variables

a) Single Request of Output Variables

When the 1-byte system variable No. 18 (12H) "FETCH_TEXT_VAR" is set on the KeTop T40, the handheld terminal automatically requests this (these) variable(s) from the PLC (**single** request) before a text with output variable(s) is displayed.

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 Variable No. 00H	b0 12H	b7 Value of var. 01H	b0		

If necessary, this automatic request of a variable can be deactivated by re-setting the system variable No. 18 (12H) "FETCH_TEXT_VAR".

After the text has been displayed, the output variables contained in the text are requested once:

	Index	Data fields					
PLC <- HT	40: Request 1-byte variable 40H	b15 Variable No. xx	b0 xx				

Following that, the PLC must continuously update the value of the variable. The telegram required for updating the 1-byte system variable is structured as follows:

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 Variable No. xx	b0 xx	b7 Value of var. xx	b0		

b) Cyclic Request of Output Variables

When the 2-byte system variable No. 2 (02H) "FETCH_TEXT_VAR_-CYCLE_TIME" is set on the KeTop T40, the handheld terminal automatically requests this variable **cyclically** from the PLC while a text with output variable(s) is displayed.

For setting and resetting this system variable, the service 15H "Send 2-byte variable" is used. If the service 15H contains a cycle time > 0 ms as value of the variable, the system variable will be set. For resetting the system variable, a cycle time of 0 ms is transmitted.

Notice
Very small cycle times increase the workload of the bus and also affect the performance of the HT.

	Index	Data fields					
PLC <- HT	15: Send 2-byte variable	b15 Variable No. b0		b15 Value of variable b0			
	15H	00H	02H	xx	xx		

According to the data type of the variable, the request is performed with one of the following services: 40H: "Request 1-byte variable", 41H: "Request 2-byte variable" or 42H: "Request 4-byte variable".

If the text to be displayed contains an output variable of data type UINT16, for example, the request telegram sent to the PLC will be structured as follows:

	Index	Data fields					
PLC <- HT	41: Request 2-byte variable	b15 Variable No. b0					
	41H	xx	xx				

Following that, the PLC sends the value of the variable to the handheld terminal using the service 15H "Send 2-byte variable".

	Index	Data fields					
PLC -> HT	15:Send 2-byte variable	b15 Variable No. b0		b15 Value of variable b0			
	15H	xx	xx	xx	xx		

When the 1-byte system variable No.22 (16H) "EVENT_VAR_NOT_ON_SCREEN" is set on the KeTop T40, the handheld terminal will be able to send an event message (EVENT_CODE) to the PLC.

The event message will be sent when a variable has been sent from the PLC to the HT, but this variable cannot be displayed any more since it has already been overwritten by another text.

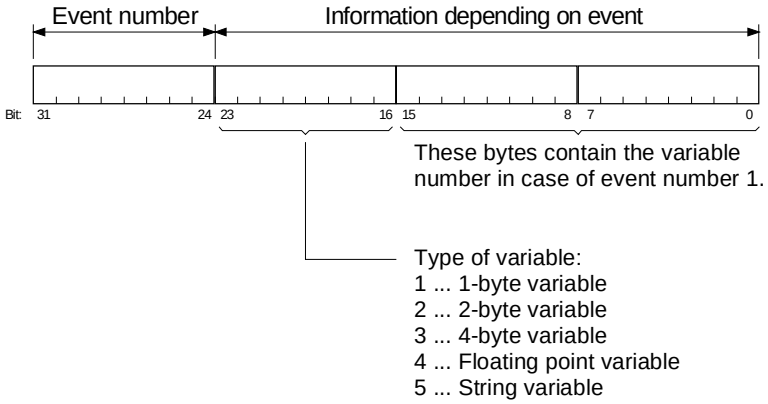
	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 Variable No. 00H	b0 16H	b7 Value of var. 01H	b0		

The automatic sending of an event message can be deactivated by resetting the system variable.

The telegram for event messages is structured as follows:

	Index	Data fields					
PLC <- HT	16: Send 4-byte variable 16H	b15 Variable No. xx	b0 xx	b31 Value of variable xx	b0 xx	xx	xx

Value of variable In this case this value contains the EVENT_CODE. The EVENT_CODE consists of 4 bytes with the following coding:

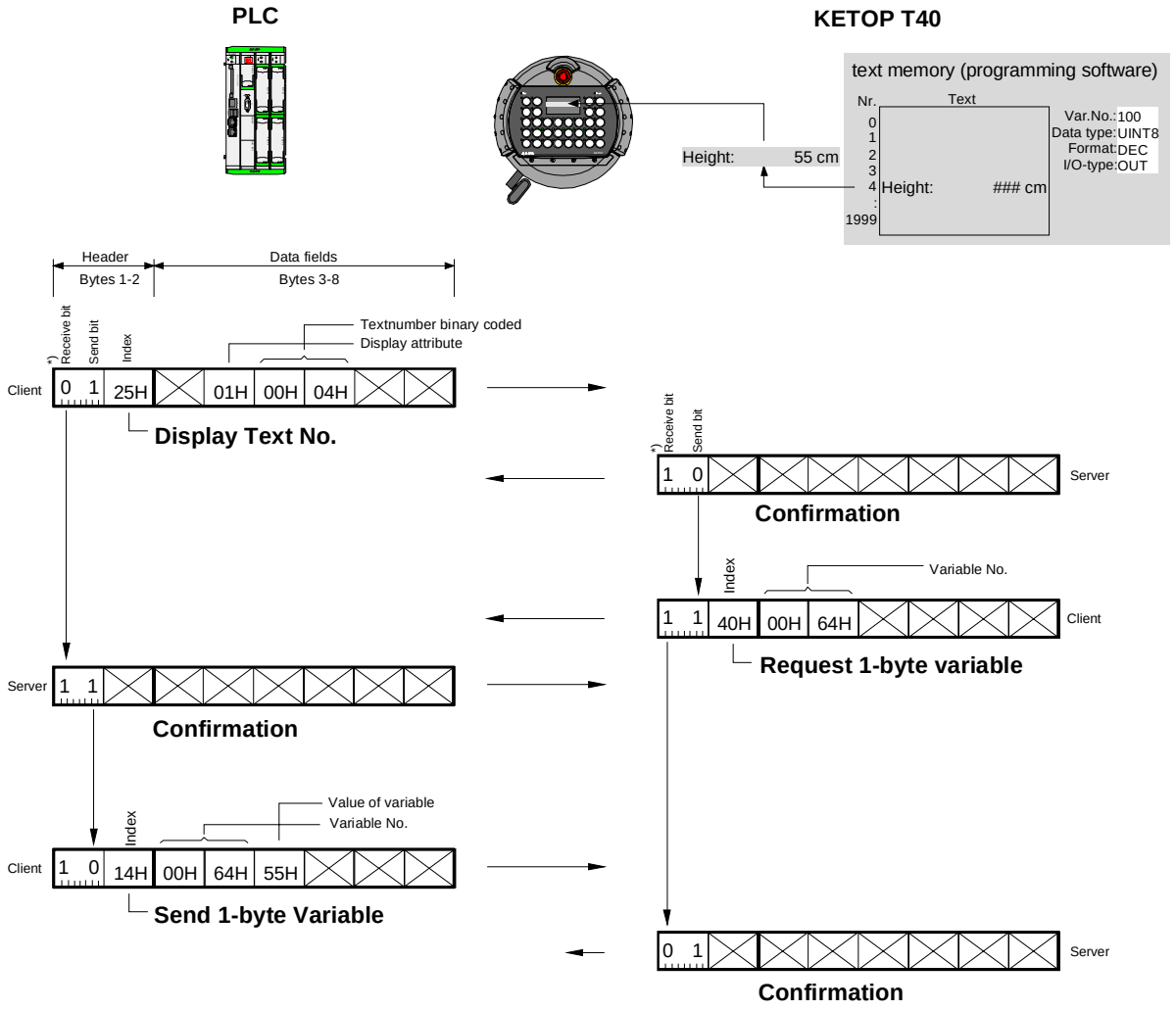


Event number 1 This event message is triggered when the PLC sends a user variable (variable number ≥ 100) to the handheld terminal where the variable should be displayed although no output field for the variable is contained in the current text mask.

Example: Displaying a Text on the KeTop T40

The PLC should activate a text display on the handheld terminal. The text contains a variable the handheld terminal requests from the PLC (once). The texts are stored in the text memory of the handheld terminal in numbered order. Therefore the PLC must send a text number to the handheld terminal. The handheld terminal reads the corresponding text from its text memory.

The text contains a variable of which the value is stored in the PLC. Before displaying the text, the handheld terminal requests the variable from the PLC which, following that, sends back a telegram including the value of the variable.
Now the handheld terminal can display the text with the value of the variable.

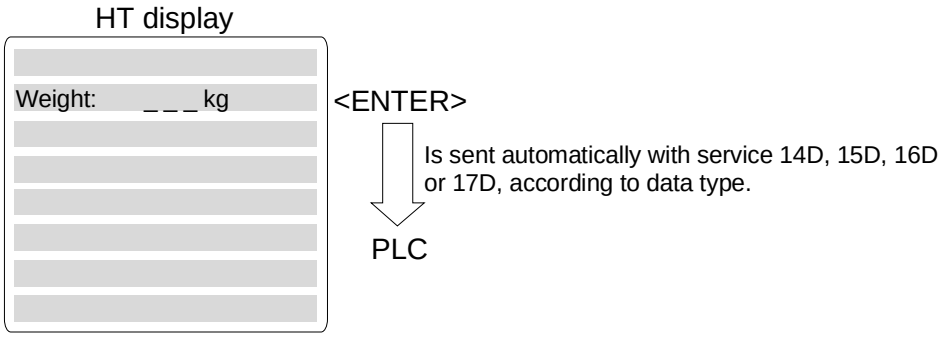


**) To get a better overview, only the receive bit and the send bit of the 1st byte of the control/status word are shown.*

Example: Displaying a text on the KeTop T40

Displaying a Text with Input Variable

The handheld terminal has already received a telegram from the PLC for displaying a text. The text to be displayed contains an input variable. The HT user types in a value and presses ENTER. Now the HT sends the value of the variable to the PLC. According to the data type of the variable, the service "Send x-byte variable" (14H, 15H, 16H) or "Send floating point" (17H) is used.

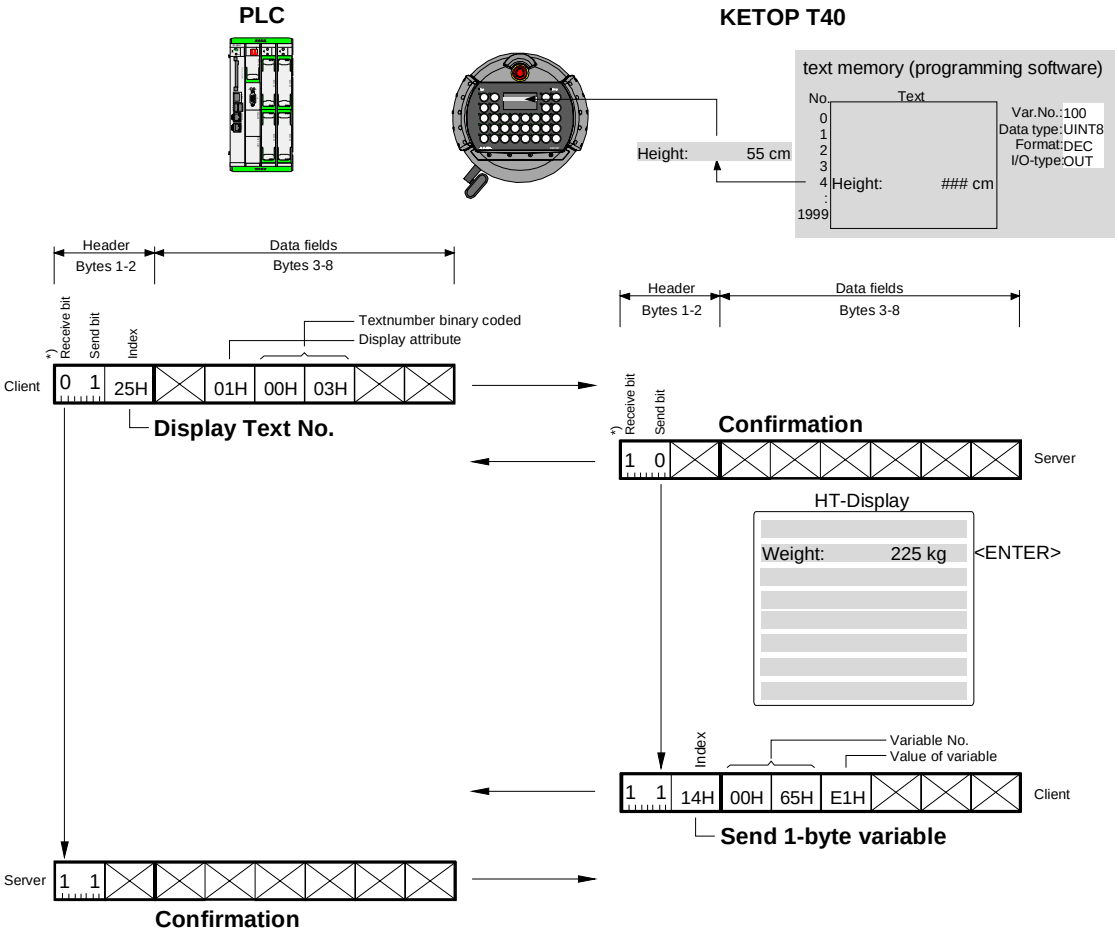


If for example the data type of the input variable is UINT8, the service 14H "Send 1-byte variable" will send the variable to the PLC.

	Index	Data fields					
PLC <- HT	14: Send 1-byte variable 14H	b15 Variable No. xx	b0 xx	b7 Value of var. xx	b0		

Example: Displaying a Text with Input Variable

The PLC should activate a text on the display of the handheld terminal. The text to be displayed contains an input variable. The user types in a value and presses ENTER to confirm the entry. Following that, the HT sends the value of the variable to the PLC.



*) To get a better overview, only the receive bit and the send bit of the 1st byte of the control/status word are shown.

Example: Displaying a Text with Input Variable

Displaying a Text with String Variable (Input or Output)

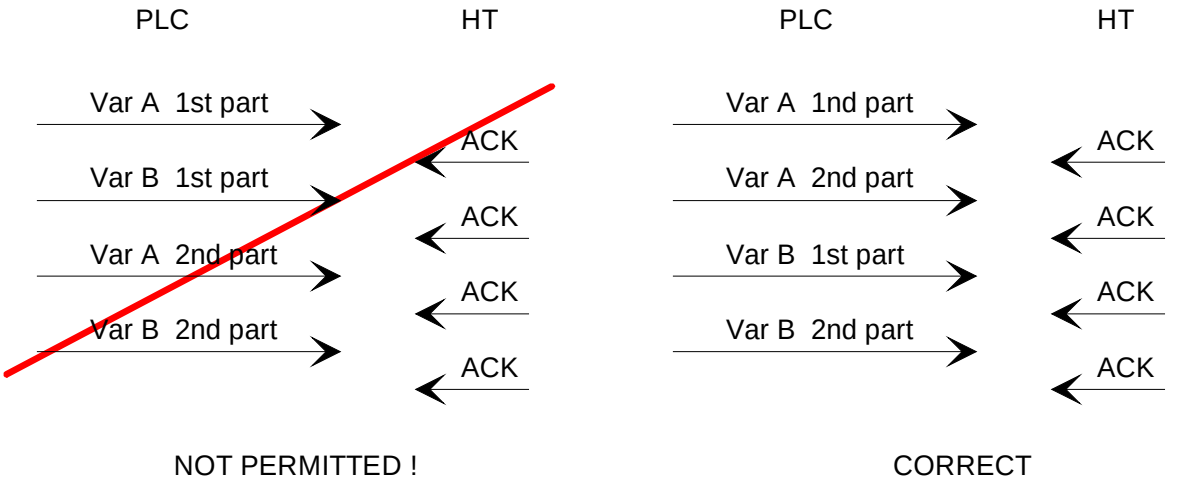
Character strings entered or to be displayed in a text are transmitted with the MMI-COM telegrams "Send bytes" and "Display bytes".

E.g. service 18H "Send bytes"

	Index	Data fields					
PLC <-> HT	18: Send bytes 18H	Length xx	Variable No. xx	Byte 1	Byte 2	...	...

Length This parameter defines the length (in bytes) of the string variable to be transmitted. If the length of the string variable exceeds the length of the available data field, the variable will be transmitted in blocks. The transmission is completed when the length of the value entered is smaller than or equal to the length of the data field. String variables can be numbered from 0 to 255.

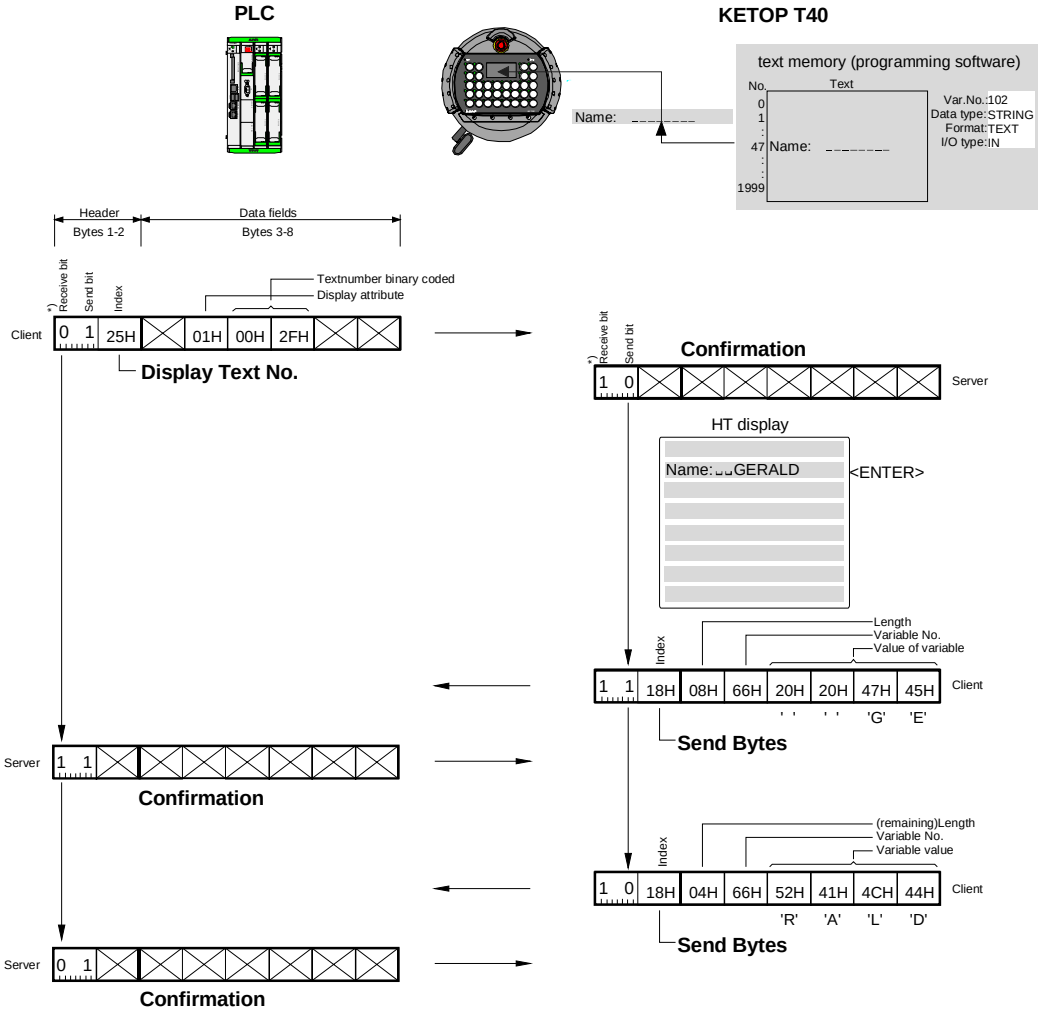
Notice
The transmission of a string variable may only start when the transmission of the previous string variable is completed (=> see below).



Correct transmission of string variables

Example: Displaying a Text with String Variable

The PLC should activate a text on the display of the handheld terminal. The text to be displayed contains a string variable. The user types in a text on the handheld terminal and presses ENTER. The handheld terminal sends the text characters entered to the PLC using the service 18H "Send bytes". If 8 characters are to be transmitted, the sending process must be performed twice.



**) To get a better overview, only the receive bit and the send bit of the 1st byte of the control/status word are shown.*

Example: Displaying a text with string variable

Text Monitor

The text monitor functions comprise two services. First the cursor must be positioned correctly on the display using the service 29H. Then the service 28H enables sending text characters displayed from this cursor position on to the HT.

Notice
A text mask with input or output fields will be overwritten when a text monitor function is executed. The cursor positions stored in the handheld terminal for the input and output fields are preserved.

=> Risk of inconsistent display.

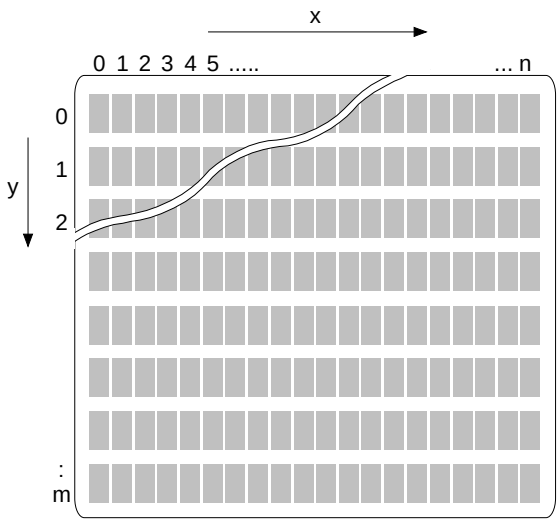
Positioning the Cursor

For positioning the cursor on the HT display, the PLC uses the service 29H. This service contains the x- and y-position of the cursor.

	Index	Data fields					
PLC -> HT	29: Position cursor 29H	Cursor x-position xx	Cursor y-position xx				

Cursor x- position/ Cursor y- position

These parameters contain the x- and y-position of the text cursor:



Displaying Text Characters from Current Cursor Position

When the cursor has been positioned on the display, the service 28H "Send text characters" enables sending a certain number of characters.

	Index	Data fields					
PLC -> HT	28: Send text characters 28H	Number of characters xx	Display attribute 01H	Text characters xx	... ...	... ...	... ...

Number of characters This parameter contains the number of characters to be transmitted.

Display attribute For displaying the text, this parameter must be set to 01H.

Text characters This parameter contains one or more characters that are displayed from the current cursor position on.

PLC activates Signal Buzzer on KeTop T40

When the 1-byte system variable No. 6 (06H) "BEEP" is set on the KeTop T40, the PLC activates the signal buzzer on the handheld terminal:

	Index	Data fields						
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	06H	b7 b0 Value of var. 01H				

The buzzer remains active until it is turned off by resetting the system variable:

	Index	Data fields						
PLC -> HT	14: Send 1-byte variable 14H	b15 b0 Variable No. 00H	06H	b7 b0 Value of var. 00H				

The buzzer on the handheld terminal can also be activated by setting the 2-byte system variable No. 3 (03H) "BEEP_TIME". In this case the duration of the beep must be preset in milliseconds:

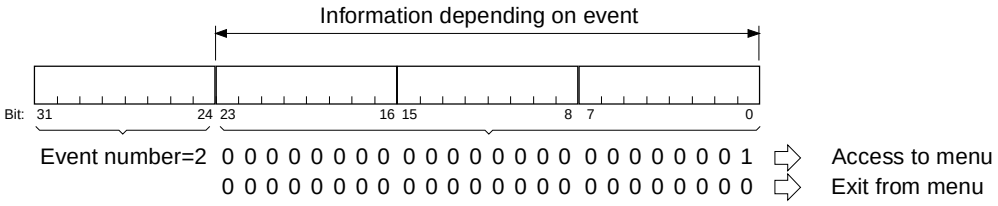
	PD-Index	Data fields						
PLC -> HT	15: Send 2-byte variable 15H	b15 b0 Variable No. 00H	03H	b15 b0 Value of variable xx	xx			

Here the value of the variable contains the duration of the beep in milliseconds.

Menu Access/Exit

When the 1-byte system variable 28 (EVENT_MENU) is set the HT sends an event message to the PLC via the 4-byte system variable 2 (EVENT_CODE) upon access to / exit from the menu.

Structure of event variable



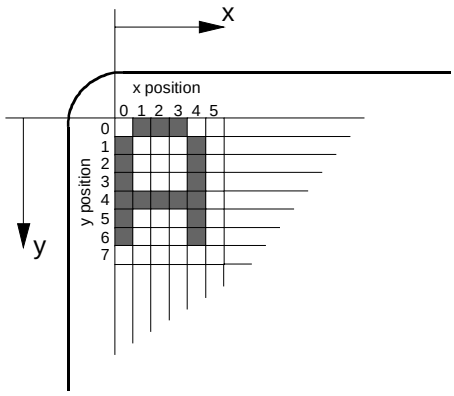
System Reset

When the value 55H is written to the system variable No. 17 (11H), a reset will be triggered on the KeTop T40. This reset corresponds to a turning on/off of the device. All values set by the PLC get lost and must be set anew.

	Index	Data fields					
PLC -> HT	14: Send 1-byte variable 14H	b15 Variable No. b0 00H	11H	b7 Value of var. b0 55H			

Graphic Functions

The graphic functions are activated from the control. The functions enable displaying a text and drawing graphical elements (line, circle, rectangle and ellipsis) at any position of the display (pixel-oriented).



Letter A in standard size at x/y position 0/0 of HT display

All graphic functions are transmitted with manufacturer-specific objects.

Graphical Elements

The elements circle, rectangle and ellipse can be represented as filled areas. To delete a graphical element, draw the same element of the same size, at the same position and in inverse color.

If you position graphical elements over existing texts or bitmaps, these texts and bitmaps will be overwritten.

Pixel

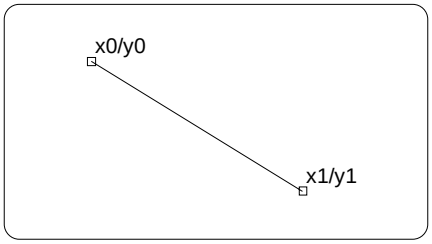
This command enables representing a single pixel at the specified x/y position of the HT display.

	Index	Data fields					
PLC -> HT	B0: Pixel B0H	x xx	y xx				

Line

This command enables representing a line on the HT display according to the specified x/y coordinates.

	Index	Data fields					
PLC -> HT	B1: Line B1H	x0 xx	y0 xx	x1 xx	y1 xx		

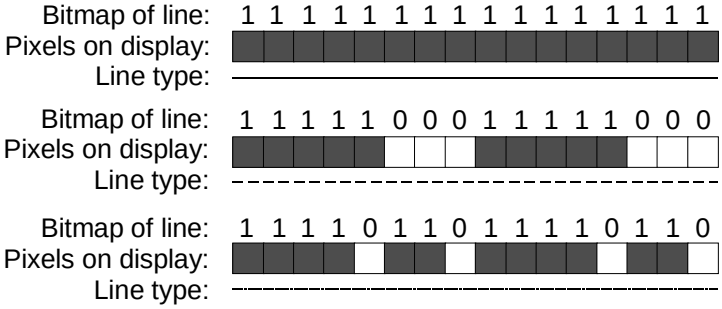


Set line type

This command is used to set a line type on which all other graphical elements are based. This setting remains valid until a new line type is defined.

	Index	Data fields					
PLC -> HT	BA: Set line type BAH	Type xx	xx				

Examples of line types:

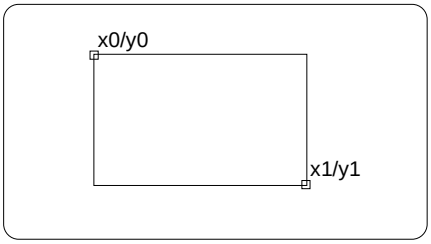


Notice
When defining a line type take care to arrange several identical bitmaps one after the other. If you choose an unsuitable bitmap the line might get irregular.

Rectangle

This command enables representing a rectangle on the HT display according to the specified x/y coordinates.

	Index	Data fields					
PLC -> HT	B2: Rectangle B2H	x0 xx	y0 xx	x1 xx	y1 xx	fill xx	



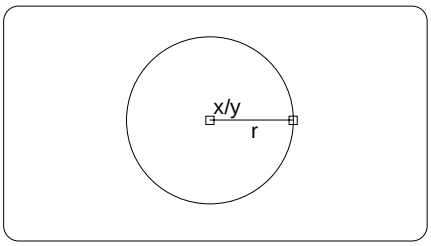
fill

0H graphic not filled
1H graphic filled

Circle

This command enables representing a circle on the HT display at the specified x/y position with the radius r.

	Index	Data fields					
PLC -> HT	B3: Circle B3H	x xx	y xx	r xx	fill xx		



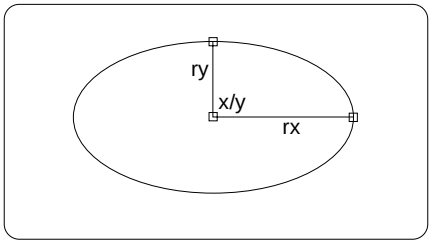
fill

0H graphic not filled
1H graphic filled

Ellipsis

This command enables representing an ellipse on the HT display at the specified x/y position with the radiuses rx and ry.

	Index	Data fields					
PLC -> HT	B4: Ellipsis B4H	x xx	y xx	rx xx	ry xx	fill xx	



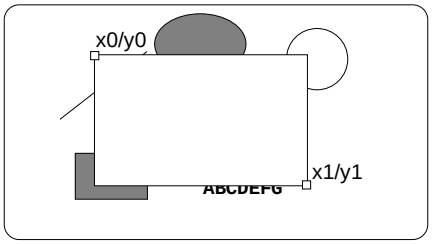
fill

0H	graphic not filled
1H	graphic filled

Clear window

This command enables clearing a rectangular area on the HT display according to the specified x/y coordinates.

	Index	Data fields					
PLC -> HT	B8: Clear Window B8H	x0 xx	y0 xx	x1 xx	y1 xx		



This command corresponds to the drawing of a filled rectangle with the set background color.

Set colors

This command enables setting the foreground and background color (at present only black or white) for all following graphical elements and graphical texts. The color remains active until another color is set.

	Index	Data fields					
PLC -> HT	B9: Set color B9H	fg xx	bg xx				

fg (foreground)

0H	white
FFH	black

bg (background)

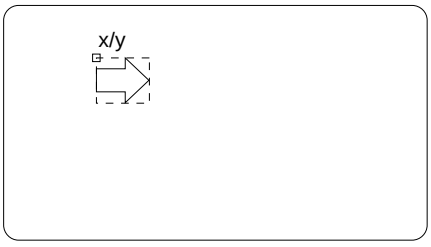
0H	white
FFH	black

Bitmaps

This command enables representing the content of a Windows bitmap file, stored in the bitmap memory of the handheld terminal, on the HT display at the specified x/y position.

The bitmap size is limited by the display size of the KeTop T40 (128x64 pixels).

	Index	Data fields					
SPS -> HT	BB: Display of BMP BBH	Bitmap- xx	No. xx	x xx	y xx		



The numbers of the bitmaps can be defined using the programming software.

Graphical Text

These texts must be prepared in the PLC and can then be sent to the handheld terminal with the indices B5 and B6.

The text can be positioned freely on the display. The text characters may be represented in double height or double width. As standard the characters are displayed in the format 5 x 7 pixels (incl. space 6 x 8 pixels). So characters with a double height have a format of 5 x 14 pixels (incl. space 6 x 16 pixels) and characters with a double width 10 x 7 pixels (incl. space 12 x 8 pixels).

Displaying a character means that the area for the character is cleared and then the character is written into the empty field. An existing graphic or bitmap representation will be deleted at this position.

Notice
A) Programmed texts stored in the HT can only be displayed in lines or columns. These texts cannot be displayed at any pixel position or with a larger size.

B) For graphical texts that are directly sent from the PLC to the handheld terminal display, no editor function is available. That means the HT will not correctly interpret symbols (___, ###) for the input and output of variables.

Text 1 / Text 2

This command enables representing a maximum of 40 text characters at the specified x/y position of the HT display (provided the characters are not displayed with a double width and begin at the pixel column 0).
The text is displayed up to the end of the line (no line folding and no continuation in the next line).

For displaying the text, at least two or more telegrams are required according to the length of the text. The first telegram (text 1) contains the position of the text on the display, the length of the text, the font size and the font attribute. The actual text is contained in one or several follow-up telegrams (text 2).

	Index	Data fields					
PLC -> HT	B5: Text 1 B5H	x xx	y xx	len xx	font_size xx	font_attr xx	

x,y Position of text on HT display

len = length: specifies number of text characters to be transmitted.

font_size = size of character: This parameter defines the size of the text characters on the HT display. The following settings are possible:

0 ... normal
1 ... double height
2 ... double width
3 ... double height, double width

font_attr This parameter contains the font attribute and specifies if the transmitted text should be displayed normally, in inverse characters, normally flashing or inverse flashing.

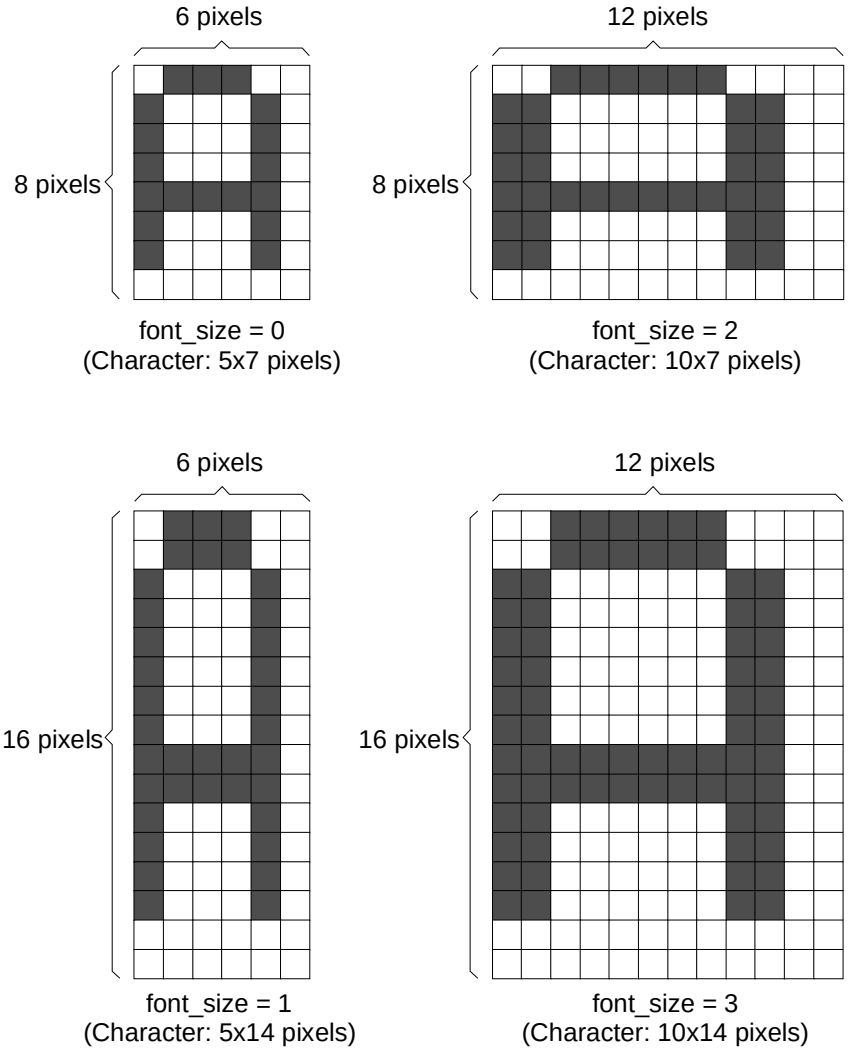
0 ... normal
1 ... inverse
2 ... normally flashing
3 ... inverse flashing

Follow-up telegram containing the text to be transmitted:

	Index	Data fields					
PLC -> HT	B6: Text 2 B6H	Text characters xx	... ...	... ...	... ...	... ...	... ...

Text characters This parameter contains one or several characters displayed from the set cursor position on (cursor position set by means of telegram „Text 1“).

Font sizes (font_size) possible on the HT display:



font_size = 0
(Character: 5x7 pixels)

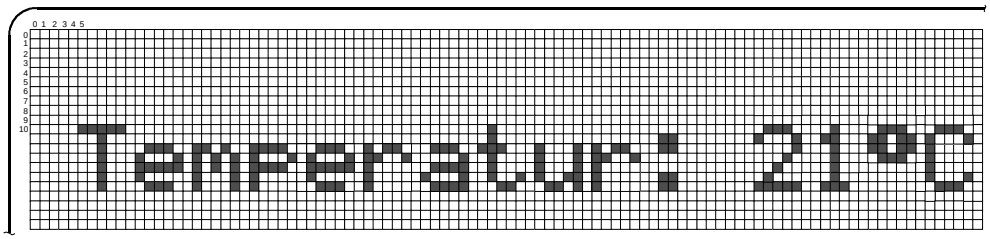
font_size = 2
(Character: 10x7 pixels)

font_size = 1
(Character: 5x14 pixels)

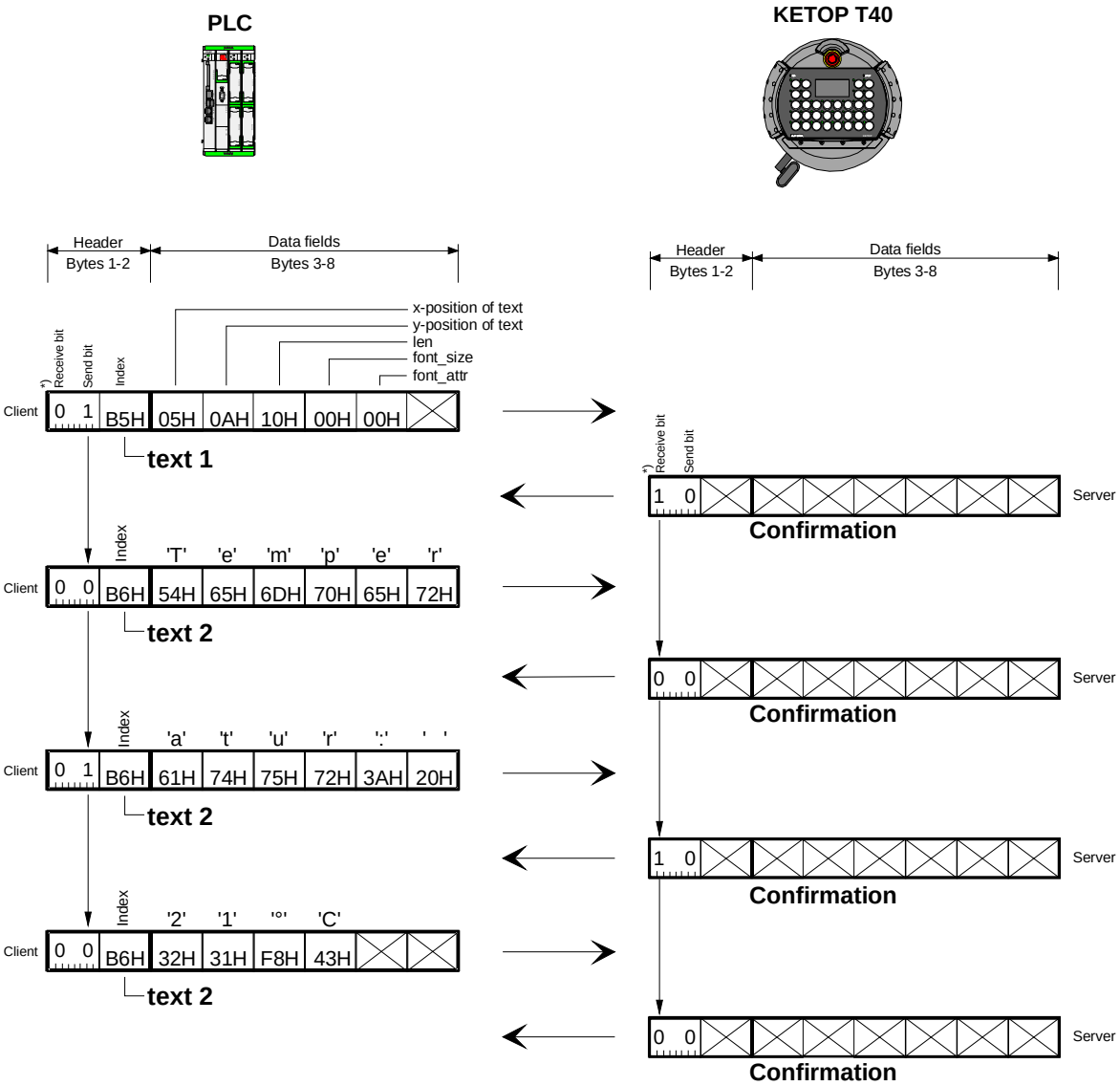
font_size = 3
(Character: 10x14 pixels)

Example of output of graphical text on HT display:

The following text should be displayed on the HT with „normal“ size at the specified position:



Exchange of telegrams:



**) To get a better overview, only the receive bit and the send bit of the 1st byte of the control/status word are shown.*